DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

Multi-objective layout optimization for material flow system with decentralized and scalable control

Multikriterielle Layout Optimierung für Materialflusssysteme mit dezentralisierter und skalierbarer Steuerung

Nikita Shchekutin
Simon Sohrt
Ludger Overmeyer

Institut für Transport- und Automatisierungstechnik
Leibniz Universität Hannover

The lack of flexibility in logistic systems currently on the market leads to the development of new innovative transportation systems. In order to find the optimal configuration of such a system depending on the current goal functions, for example minimization of transport times and maximization of throughput, various mathematical methods of multi-criteria optimization are applicable. In this work, the concept of a complex transportation system with decentralized and scalable control is presented. Furthermore, the question of finding the optimal configuration of such a system through mathematical methods of optimization is considered.

[Keywords: logistics, material flow system, matrix conveyor, warehousing]

Der Mangel an Flexibilität bei heutigen Logistiksystemen führt zur Entwicklung von neuartigen Fördersystemen. Um die optimale Konfiguration der Systeme anhand aktueller Zielfunktionen, wie beispielsweise Minimierung von Transportzeiten und Maximierung des Durchsatzes, herauszufinden, sind verschiedene mathematische Ansätze zur simultanen Optimierung verschiedener Kriterien anwendbar. In dieser Arbeit wird das Konzept für ein komplexes Materialflusssystem mit dezentraler und skalierbarer Steuerung vorgestellt. Außerdem wird die Frage nach einer Konfigurierung solcher Systeme durch mathematische Optimierungsmethoden betrachtet.

[Schlüsselwörter: Logistik, Materialflusssysteme, Fördermatrix, Intralogistik]

1 INTRODUCTION

Existing methods of industrial facility optimization are concentrated on arranging the production machines inside the factory plant. While main objective value is the minimization of Material Handling Cost (MHC), less or even no attention is paid to how exactly a material flow between single machines is implemented. It is just said, that continuous (belt conveyors) or discontinuous conveyors (forklift trucks) might be used [ACP17, KG13].

However, recent developments in the area of material flow systems make the material flow arrangement task more complex. The trend of modularization and small-scale of material flow systems significantly increases the solution space. Several research institutes and industry companies have developed flexible material flow systems with the size of the transport module, smaller than the conveyed goods. Some of them are presented in the Figure 1.

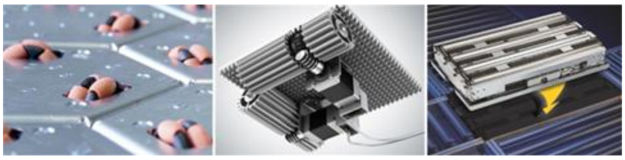

Figure 1. Small-scaled modular conveyor systems: Celluveyor [UTF10, UTF16], Motion Cube [Fes15] and Magic Carpet [Ito16]

In this work the question of arrangement of such material flow system knowing the positions of entry, exit and equipment taking into account the current state of the art will be considered.

The paper has the following structure. Firstly, the current state of the art in the area of facility optimizations and material flow systems will be given. Afterwards two steps for a material flow system arrangement will be proposed and realized: optimization methods will be implemented and their results will be presented and discussed.

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

2 Related Research

The question of optimal arrangement of a given number of facilities within a given space was firstly formulated in [KB57]. This was termed the Facility Layout Problem. Later, many different formulations and solution approaches were proposed. They will be shortly overviewed in the next subchapter based on the latest surveys in the area [AV17, DPH07, SS06].

2.1 Facility Layout Problem

The complexity of FLP as any optimization problem depends on its formulation and assumptions. The formulation approaches can be basically classified into simplified and complex (Table 1).

Table 1. Formulations of FLP [AV17, DPH07, SS06]

Formulation	Location	Parameters	Layout	Objective Function
Simplified	Discrete [CZF17, KB57]	Static	Regular	Single-objective
Complex	Continual [HDV16]	Dynamic [CR16, DRW05]	Irregular	Multi-objective [KS17, Mat15, MSM13]

The different aspects can be combined; each complex parameter makes the task formulation more practice-related, but increases the parameter space and computational requirement. For example, the simplest formulation consists of dividing the planning area in multiple predefined blocks (discretization), where n facilities have to be put in n location. Objective values don't change with time (staying static), all the objects have a quadrilateral shape (or at least regular), the only objective value is to minimize the Material Handling Cost (MHC) and the constraint is assigning each location just one facility. In this case objective function has a quadratic dependence on variables and the constraints are linear – such formulation is called Quadratic Assignment Problem.

The inherent objective value for FLP is minimization of the MHC, which is often simplified to the minimization of the distance between the machines. The distance is said to be covered with discontinuous conveyors (forklift trucks, AGV) without any further details. However, recent trends of intellectualization of production systems, modularization and integration of material flow in production cycles [GF11, SHE17] show the need of considering an arrangement of a material flow system as a separate step.

In the following subchapter an example of an intelligent production system will be presented shortly. It was developed at the ITA within the framework of netkoPs research project [ITA17] and is called netkoPs.Lab.

2.2 netkoPs.Lab – Intelligent Production System

In netkoPs.Lab the material flow is performed with the help of continuous conveyors of two types – belt conveyors and novel small-scale conveyor modules [ITA13, Krü15, KSO16]. These modules can be combined in conveyor matrices of different shapes and sizes by plug-and-play concept due to the modular control and communication systems. (Figure 2)

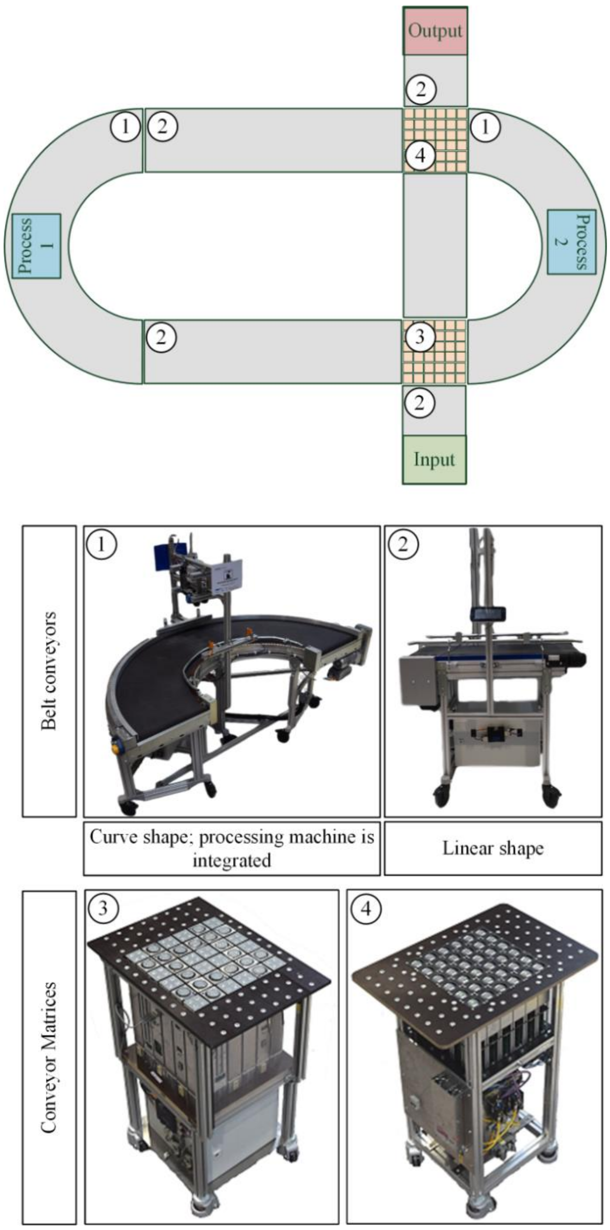

Figure 2. netkoPs.Lab

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

The size of these modules is much smaller than transported packets, which allows a wider range of logistic functions. Due to the current high production costs of single modules, they have to be combined with the belt conveyors. Apart from conveyor systems, processing machines, monitoring systems and data storage are also integrated into the system. The united description language ProductionML was developed [SHO15]. Coming into the system the packet gets its route calculated and reserved in order to avoid deadlocks.

One possible layout of netkoPs.Lab is presented in the Figure 2. It consists of two conveyor matrices, seven belt conveyors, two processing machines, and an identification system. The modularity of the system elements makes layouts easily adaptable. However, reconfiguration planning still requires cost and effort. Product individualization requires calculation of the optimal configuration of a material flow system to occur automatically, simultaneously meeting changing requirements. Determination of the layout of such a system is not possible without taking into account the available material flow elements.

In this work, the discrete static regular multi-criteria formulation will be considered (Table 1). Sample layout of netkotPs.Lab in this case can look as in Figure 3. Discrete unit size is equal to the size of single conveyor module.

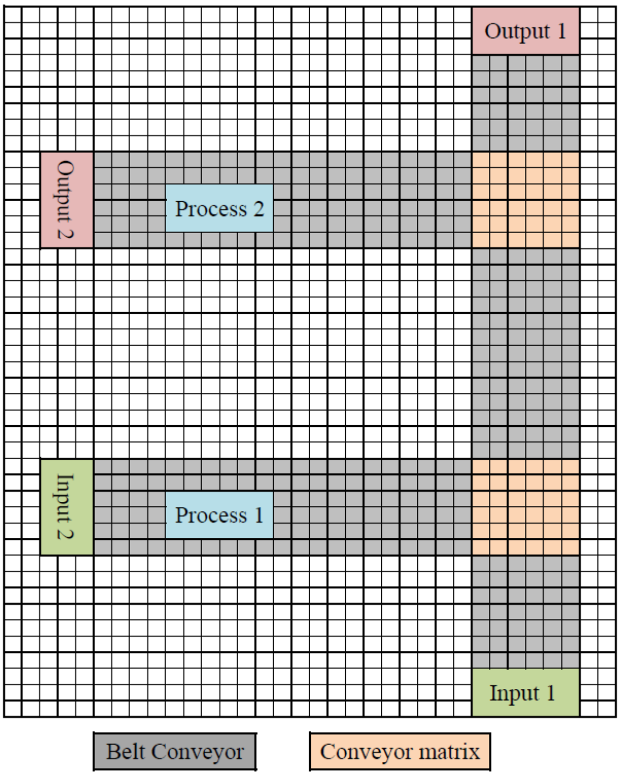

Figure 3. Discrete layout of netkoPs.Lab

In the work the positions of inputs, outputs, processing machines (processes), sizes and destinations of packets as well as the number and sizes of available belt conveyors and matrices are predefined (Figure 4). The task is to find the positions of conveyors taking into account the constrains and satisfying the objective values.

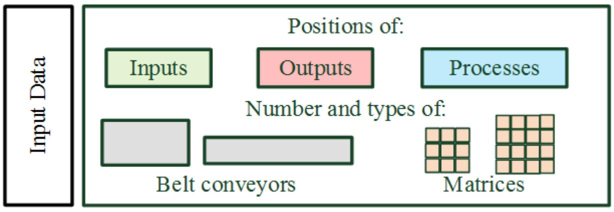

Figure 4. Input data for the optimization algorithm

The objective values for the layout optimization will be reviewed in the next subchapter.

2.3 OBJECTIVE VALUES FOR LAYOUT OPTIMIZATION

All the objective values can be divided into quantitative and qualitative. Quantitative objective values are:

- Min. Material Handling Cost/ haul capacity
- Min. Stock holding costs
- Min. Reconfiguration and adjustment costs
- Min. occupancy costs
- Min. Time of storage
- Min. throughput time

Qualitative objective values are:

- Max. Observance of Production
- Max. On-the-job safety
- Max. elasticity
- Max. flexibility

Minimization of MHC is the most common objective value, though its mathematical formulation differs. In some cases it is assumed to be just the distance between single destinations. In more complex approaches, it is summed from such parameters as costs for personnel, conveyors, electricity, external services etc.

Some objective values make sense only in complex formulation. For example, minimization of reconfiguration and adjustment cost occur only with dynamic objective parameters. An important question there is if it makes sense to invest in reconfiguration to better achieve the values or proceed with the existing ones.

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

Results of optimization depend on the formulation approach, although multiple optimization criteria must be taken into account to consider practice-related scenarios. In the following subchapter state of the art in multi-objective optimization will be presented.

2.4 Multi-objective combinatorial optimization

One common method to address the multi-objective Facility Layout Problem (MOFLP) is to formulate a single-objective optimization problem by using a scalarized combination of real-valued objective functions of the initial MOFLP. This is often done with the help of the weighted sum method. In this case two separate tasks have to be solved: choosing the weights and quantifying the qualitative factors.

Often it is addressed by involving the decision maker in the process (generate several layouts and let user choose one; collect the user's preferences first then generate the single layout; weight the single objective values using expert's experience). A disadvantage in this case is the absence of universal approaches, as just a user-dependent solution is received [Mal89]. Another variant consists of the application of heuristics to weighting aspects [CS05, Mat15, MG07, SS10]. Thus, in [SS10] it was proposed to use a 3-stage approach, which consists of input data normalization, determination of relative weights, and generation of layouts by 4 different algorithms. The layout with the minimal objective function is finally chosen.

To make locating a single optimal solution completely without human guidance one can apply several decision-making techniques from the area of multi-criteria design and evaluation problems. In [ALE13] the Electre method is successfully applied, which allows evaluation of alternatives to also take into account the uncertainty and ambiguity of performance. A common alternative to weighted sum method is the ε-constraint method, which resolves the problem by minimizing one objective, while others are transformed to constraints. Further approaches are the elastic constraint method, Benson's method, and the achievement function method [Ehr05].

Scaling methods suit Pareto class formulations of FLP, but often the conflicting objectives have to be considered in a hierarchical manner, when the objective values have predefined priorities (law prescription etc.) Such optimization problems are called lexicographic. Another type of problems consists of the so-called max-ordering problems, which are often applied to optimal location planning with the goal of minimizing the distance between located instances [OWM08]. In this case non-scalarazing approaches as distributed or multi-agent optimization can be applied, where each agent generates optimal decisions based on its own cost function and exchanges these outcomes with its neighbors. [LCW17, NO09]

Combinatorial optimization is based on searching the optimal combination of a defined number of elements. Common optimization algorithms are briefly presented in the next subchapter.

2.5 Optimization algorithm

There are several optimization algorithms, which can be classified into three types: exact, evolution and local search algorithms (Table 2).

Table 2. Optimization algorithms and their classification

Exact Algorithms	Evolution Algorithms	Local search algorithms
Branch-and-bound Cutting plane Blind search	Genetic Algorithms Ant Colony algorithm	Tabu Search Simulated Annealing

Exact algorithms are easier to implement, but they are only applicable to rather small-sized problems. If the task becomes NP-Hard, then local search algorithms and evolution algorithms must be applied. Tabu Search and Simulated Annealing are part of the class of local search algorithms [CK96, CBM98, SZO16]. These algorithms are based on an iterative transition from one feasible solution to another, with a consequent evaluation of each solution, until a stopping rule is satisfied. For example, there are used in the case of a system that is not improved through a defined number of variations. For such methods, often the heuristics of sticking into a local optimum are required.

Apart from the local search algorithms, evolution algorithms have also proved their applicability to the described problem. By these methods, a number of solution candidates are generated for the first population and for each of these candidates there is an evaluation of its performance with regards to the existing objective. Afterwards the candidates that perform best during the evaluation will be selected and combined through mutation, and a more evolved population will be generated. Therefore, new solution candidates will be received even though diverse implementation problems often occur. For instance, as a result of the use of genetic algorithms the problem of programming candidate solutions occurs. [DRW05, PCP03]

The Ant Colony algorithm is also a part of evolution algorithms; it proved its applicability in unconstrained or single parameter constrained cases, which may underline

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

the general problem of all evolution algorithms. [BDS06, SVS05]

The important question for all combinatorial optimization algorithms is the generation of initial solution. In this work path-finding algorithm will be used for finding initial non-integer solution.

2.6 PATH-FINDING

The task of path-finding algorithms is to find the shortest route between two points. All the positions of entry, exit and processing points are known in advance. This divides the task into two stages: searching the paths and covering them with the available material flow elements.

The typical algorithm for path-finding is the A* algorithm [HPK17]. It starts with the weighted graph, which can be created by discretization of an area into a grid. Afterwards the single grid units are evaluated on their distance to the destination point. The evaluation function for A* algorithm is:

$\boldsymbol{f}(\boldsymbol{n}) = \boldsymbol{g}(\boldsymbol{n}) + \boldsymbol{h}'(\boldsymbol{n})$, *where g(n) is a real cost of getting to the point n and* $\boldsymbol{h}'(\boldsymbol{n})$ *the heuristic value of transport from the node n to the goal node*

In chapter 3 the detailed example of evaluated grid will be given.

3 FORMULATION OF THE PROBLEM

In this work the task of multi-objective material flow optimization will be solved with two steps: the initial solution through path-finding and combinatorial optimization using branch-and-bound algorithm for covering the path with available conveyors. The approaches will be described in further, after the objective function is defined.

3.1.1 WEIGHED OBJECTIVE FUNCTION

The primary goal of this work is to initiate the multi-objective optimization approach and test its boundaries with an exact algorithm. For this purpose there will be two objective values formatting the objective function – one quantitative (cost) and one qualitative (flexibility). Quantification of flexibility in the simplest way will be performed by a reward – with the use of each conveyor matrix the objective value will be reduced by 1. Thus, the resulting objective function will look as following:

$$\min f(x) = \left(\alpha * \left(\sum_i c_i * n_i + cm * n_m \right) - \beta * (n_m * p) \right),$$

$$s.t. \alpha + \beta = 1$$

$$where:$$

α, β – *weight coefficients (cost and flexibility).*

n_b, n_m – *number of belt conveyors and conveyor matrices*

c_i, cm – *costs of belt conveyors and conveyor matrices*

p – *reward value*

The definition of weight coefficients will be left to the experienced user. Price is calculated in the conditional units. All the inputs must be connected with all outputs and the processing points are not taken into account.

3.2 PATH-FINDING: MATHEMATICAL FORMULATION OF OVERLAPS

The methods are concentrated just on searching for the optimal paths; subsequent steps are not considered. Applying them to the material flow, overlapping of routes occurs. This will be shown in a further example with the path-finding algorithms implemented in Matlab.

The example has following parameters: the grid of 20x14 cells, task is to transport 3x3 packets from inputs to outputs. The found path in marked with blue and orange arrows in Figure 5.

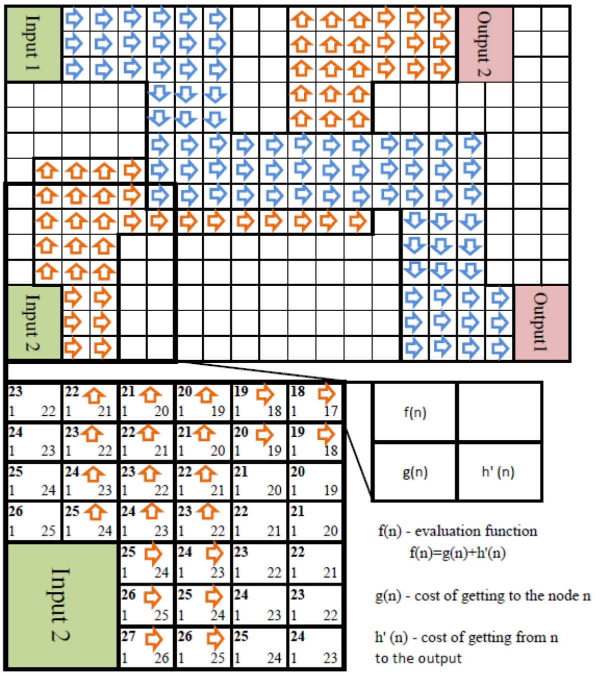

Figure 5. The case of partial overlaps.

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

After the evaluation function of each node is calculated, the path can be found. Each node gets three values depicted: $\boldsymbol{g}(\boldsymbol{n})$ is presented at the left down corner, $\boldsymbol{h}'(\boldsymbol{n})$ is placed at the right down corner and $\boldsymbol{f}(\boldsymbol{n})$ is shown at the upper left corner. It can be seen that the paths have partial overlaps. This leads to extra costs in the case that these paths will be covered with conveyors – they would be wider than necessary.

In a perfect case full superposition would be received, but more often partial superposition happens. To avoid partial superposition the same path-finding algorithm must be run a few more times, each time with some extra conditions with unusable points defined. These are the points, which can't be crossed by a secondary route.

In this work it is assumed that during the route calculation only 90 degree turns are allowed (1-2-3, see Figure 6, left). This leads to the definition of unusable points, which is presented in Figure 6 , right.

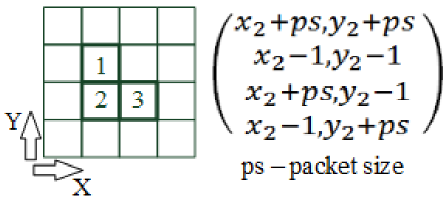

Figure 6. Definition of coordinates for unusable points

After the full routes superposition is reached (Figure 7), the system must be arranged with the material flow elements. For the purpose the Branch-and-Bound algorithm will be used. After the path is found, the methods of combinatorial optimization must be used to choose the appropriate conveyors for covering the path.

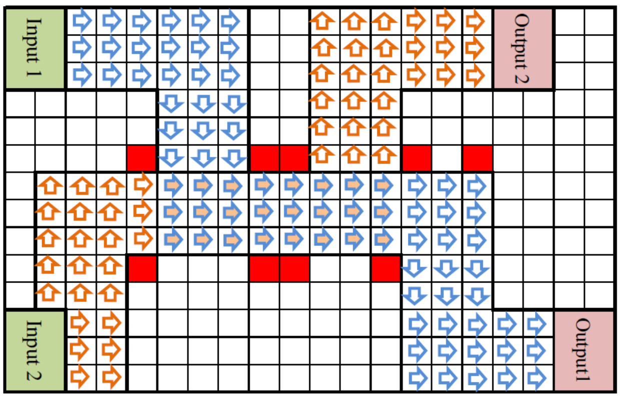

Figure 7. Found paths with unusable points defined

3.2.1 BRANCH-AND-BOUND ALGORITHM

This algorithm is a typical non-heuristic solution for small-sized combinatorial NP-Hard problems. The pseudocode for the algorithm in the work is presented and explained further.

Branch-and-bound: Layout optimization

Input: *Coordinates of Inputs and Outputs, number and length of belt conveyors, number of conveyor matrices*

Set $L = \{X\}$; initial non-integer solution $\hat{x}$

Branching $L \rightarrow S(x: integer)$

while $L \neq 0$

| Select a sub-problem S

| **if** $\widehat{x'}$ can be found and f($\widehat{x'}$)<f($\hat{x}$):

| | Set $\hat{x}$=$\widehat{x'}$

| | S→ $subbranches$

| | Insert the $subbranches$ into L

| Remove S from L

Return $\hat{x}$

Output: $\hat{x}$: *Coordinates of conveyors*

Firstly for the main problem a relaxation to non-integer constraints is performed. Then the found decision is checked to see, if variables with integer constraints have integer values. In the rare cases it is so – then the problem is solved. If not, then two sub-problems (branches) are created, where the first real-valued variable with integer constrains gets respectively the next bigger integer value and the previous smaller integer value assigned. If the objective values of the branch are worse than existing/initial ones, then this branch does not need to be considered anymore. In opposite case, the solution is saved as a current optimum. Then the next real-valued variable get the integer value, creating further branches and the evaluation is repeated. At the end the full systematic enumeration of feasible solution space will be received. [MJS16]

At the beginning the paths are covered with conveyors of random real-valued size and conveyor matrices. Then two first branches are created: the first belt conveyor is substituted with the closest longer and shorter ones from the given set. The optimization performed again and objective function is evaluated for the both cases. The best one is divided into two branches by choosing two integer values for the second conveyor. This process goes on till the distance is covered with the belt conveyors and matrices from the given set.

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

The complete optimization process looks as presented in Figure 8. Path-finding and Branch-and-Bound will be implemented in Matlab.

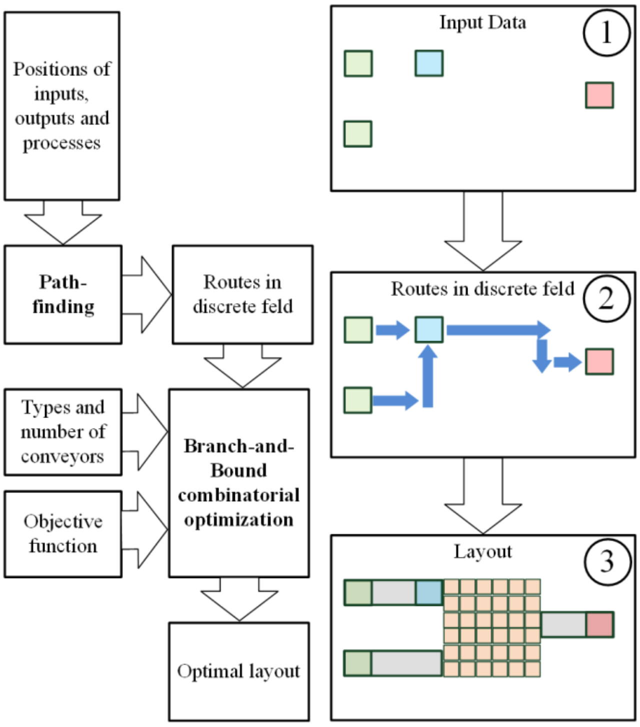

Figure 8. Sequence of the whole optimization process

4 RESULTS

4.1.1 STEP 1. PATH-FINDING

The user defines the size of the working space, number of inputs, processing points and types of packets (outputs). For the project the size of the packets stays equal – 3x3. After that the user defines the positions of inputs, processing points, and outputs. The positions in the example of the work are presented in Figure 9. In the example 4 inputs, 4 outputs and 4 processes are defined. Following routes are set:

- (I1/I2/I3/I4) – P1 – O1
- (I1/I2/I3/I4) – P2 – O2
- (I1/I2/I3/I4) – P3 – O3
- (I1/I2/I3/I4) – P4 – O4

Firstly, A* algorithm is used. Found paths are presented in Figure 9.

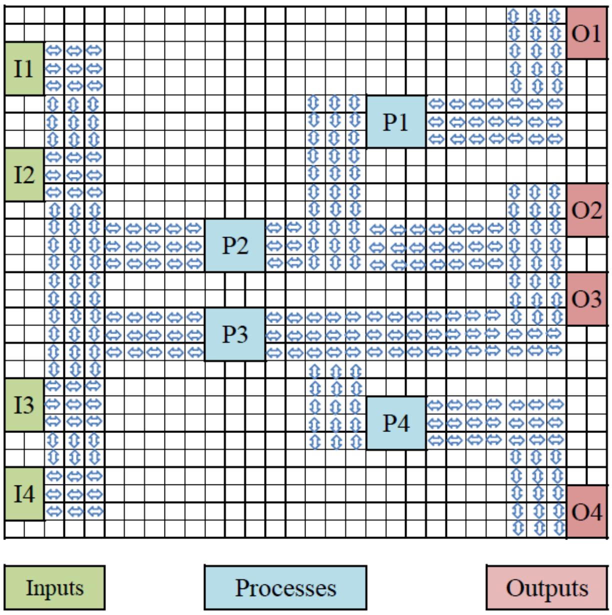

Figure 9. Results of A (left) and Dijkstra Algorithms: Rectangular routes.*

4.1.2 STEP 2. BRANCH-AND-BOUND

Flexibility is a qualitative value, which needs to be quantified. In this work it will be done by using the reward value. For the each conveyor matrix used, the objective function will be deducted by 1. In our case defining the weight coefficients is left to the user. The user also has to define the positions of inputs and outputs and lengths and number of available conveyors.

The solution for 0.8 (cost weight) and 0.2 (flexibility weight) coefficients is presented in Figure 10. This is the simplest case of multi-objective optimization, which is why changing of the second parameter directly influences the number of conveyor matrices in the system. The solution for 0.35 (cost weight) and 0.65 (flexibility weight) coefficients is presented in Figure 12.

The performance (number of iterations) of the branch-and-bound algorithm depends on the size of the problem. Important parameters are: size of the area, number of inputs and outputs, and number and sizes of available material flow elements.

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987

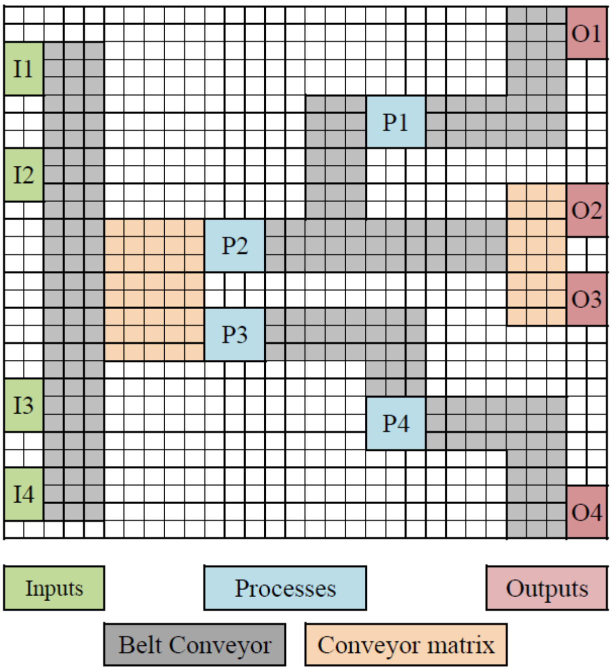

Figure 10. Layout: cost weight α =0.8, flexibility weight β =0,2.

The number of iterations grows exponentially with the number of outputs. So the task for 4 Inputs and 7 Outputs could not be solved. This becomes even more complex if the planning area is doubled. For some configurations the solutions could not even be determined (Figure 11).

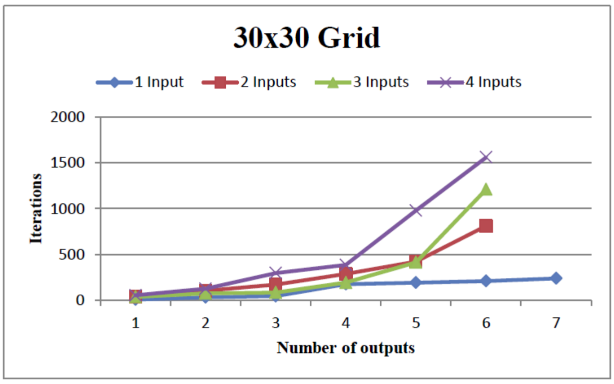

Figure 11. Number of iterations for 30x30 grid

5 Conclusion and Outlook

In this work the optimization of a material flow system as a subsequent step of FLP was considered. After an overview of related research, two-step solution approach was proposed and implemented in Matlab – path-finding and multi-objective branch-and-bound optimization.

As an approach to a multi-objective optimization, weighting was implemented. Including the user into the weighting made the solutions user-dependent. Weighting is a typical scaling method, which is easy to implement, but hard to make universal.

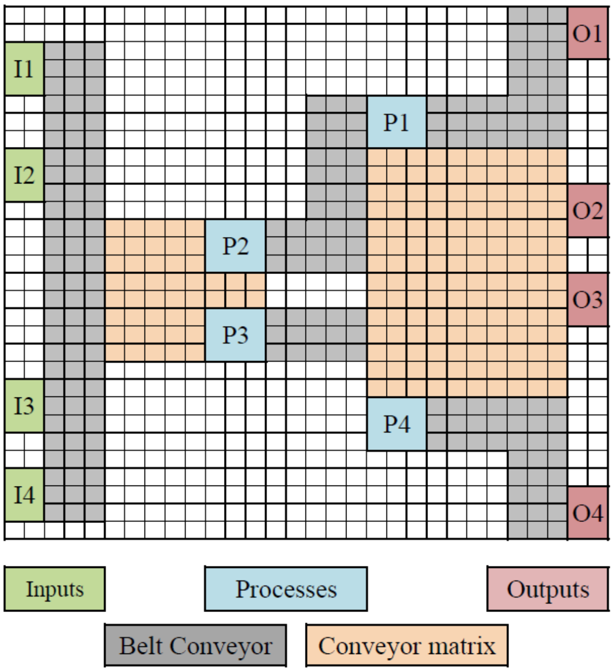

Figure 12. Layout: cost weight α =0.35, flexibility weight β =0,65.

The implemented A* path-finding algorithms can be optimized. The need of full-superposition forces the algorithm to run multiple times, subsequently adding new unusable points. Instead of that the evaluation function could be adjusted by adding an extra parameter, which would expand the evaluation function, if there is already a path coming through this node. In this case the complete evaluation function would look like this:

$\boldsymbol{f(n) = g(n) + h'(n) + s(n)}$,*where s(n) – is a penalty value which comes for every secondary path going through the node n.*

Branch-and-bound showed its applicability to the smallest problem sizes. At maximum 4x4x6 tasks could be solved, when the field was a 30x30 grid and packet size was limited to a 3x3 units. This is explained by the weak abilities of the algorithm itself as well as by the hard integer constraints.

In this work the arranging of a material flow system was taken as a subsequent step to solve the FLP. The positions of entry, exits, and processing machines are defined without considering the available material flow resources. The reformulation of FLP would make more sense, when all the system elements already would be located optimally for material flow system. This would add extra constraints to any definition of FLP.

Future research is planned to reformulate the FLP with accent on a material flow. Moreover, further approaches to multi-objective optimization will be implemented. Multi-objective function will be divided into subfunctions and for each a separate optimization routine will be implemented. The results will be presented in the way of non-conflicting Pareto-front.

Currently additional conveyor matrices are being built and integrated into the netkoPs.Lab demonstrator. In the future layout solutions will be validated there.

6 Literature

[ALE13] Aiello, G.; La Scalia G.; Enea, M. (2013): *A non-dominated ranking Multi Objective Genetic Algorithm and Electre method for unequal area facility layout problems*. In: Expert Systems with Applications 40/12(12):4812–4819

[AV17] Anjos, M. F.; Vieira, M. V. (2017): M*athematical Optimization Approaches for Facility Layout Problems: The State-of-the-Art and Future Research Directions.* In: European Journal of Operational Research

[ACP17] Azevedo, M. M.; Crispim, J. A.; Pinho de Sousa, J. (2017): *A dynamic multi-objective approach for the reconfigurable multi-facility layout problem*. In: Journal of Manufacturing Systems 42:140–152

[BDS06] Baykasoglu, A.; Dereli, T.; Sabuncu, I. (2006): *An ant colony algorithm for solving budget constrained and unconstrained dynamic facility layout problems*. In: Omega 34/4(4):385–396

[CR16] Chae, J.; Regan, A. C. (2016): *Layout design problems with heterogeneous area constraints*. In: Computers & Industrial Engineering 102:198–207

[CZF17] Che, A.; Zhang, Y.; Feng, J. (2017): *Bi-objective optimization for multi-floor facility layout problem with fixed inner configuration and room adjacency constraints*. In: Computers and Industrial Engineering 105:265–276

[CS05] Chen, C.W.; Sha, D.Y. (2005): *Heuristic approach for solving the multi-objective facility layout problem*. In: International Journal of Production Research 43/21(21):4493–4507

[CK96] Chiang, W.C.; Kouvelis, P. (1996): *An improved Tabu search heuristic for solving facility layout design problems*. In: International Journal of Production Research 34/9(9):2565–2585

[CBM98] Chwif, L.; Barretto, M.,R.; Moscato, L. A. (1998): *A solution to the facility layout problem using simulated annealing*. In: Computers in Industry 36/1(1):125–132

[DPH07] Drira, A.; Pierreval, H.; Hajri-Gabouj, S. (2007): *Facility layout problems: A survey*. In: Annual Reviews in Control 31/2(2):255–267

[DRW05] Dunker, T.; Radons, G.; Westkämper, E. (2005): *Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem*. In: European Journal of Operational Research 165/1(1):55–69

[Ehr05] Ehrgott, M. (2005): *Multicriteria optimization*. 2nd ed. Springer, Berlin, Heidelberg, New York

[Fes15] Festo GmbH Festo (2015): *Palettiersystem MotionCube*. http://www.handling.de/video---festo---palettiersystem-motion-cube.htm

[GF11] Gue, K. R.; Furmans, K. (2011): *Decentralized control in a grid-based storage system*. In: 61st Annual IIE Conference and Expo Proceedings

[HDV16] Huchette, J.; Dey, S. S.; Vielma, J.,P. (2016): *Strong mixed-integer formulations for the floor layout problem.* In: Information Systems and Operational Research. ISSN: 0315-5986

[HPK17] Hudziak, M.; Pozniak-Koszalka, I.; Koszalka, L.; Kasprzak, A. (2017): *Multi-agent pathfinding in the crowded environment with obstacles: Algorithms and experimentation system.* In: Journal of Intelligent and Fuzzy Systems. 32/2(2):1561–1573.

[ITA17] Institut für Transport- und Automatisierungstechnik: *Vernetzte, kognitive Produktionssysteme(netkoPs)*. http://www.netkops.de/.

[ITA13] Institut für Transport- und Automatisierungstechnik (2013): *CogniLog System*. http://www.cognilog.uni-hannover.de/

[Ito16] Itoh Denki Zweigniederlassung Deutschland (2016): *Immer zum richtigen Ausgang schleusen*. http://www.materialfluss.de/lager-und-kommissioniertechnik/immer-zum-richtigen-ausgang-schleusen/

[KB57] Koopmans, T. C.; Beckmann, M. (1957): *Assignment problems and the location of economic activities*. In: Econometrica journal of the Econometric Society, an international society for the advancement of economic theory in its relation to statistics and mathematics 25/1(1):53–76.

[KG13] Kothari, R.; Ghosh, D. (2013): *Tabu search for the single row facility layout problem using exhaustive 2-opt and insertion neighborhoods*. In: European journal of operational research. EJOR 224/1(1):93–100.

[Krü15] Krühn, T. (2015): *Dezentrale, verteilte Steuerung flächiger Fördersysteme für den innerbetrieblichen Materialfluss*. Zugl.: Hannover, Univ., Diss., PZH-Verl.; TEWISS - Technik-und-Wissen-GmbH, Garbsen, Garbsen

[KSO16] Krühn, T., Sohrt, S., Overmeyer, L. (2016): *Mechanical feasibility and decentralized control algorithms of small-scale, multi-directional transport modules*. In: Logistics research 9/1(1)

[KS17] Kumar, R.; Singh, S.P. (2017): *Optimal selection of multi-criteria unequal area facility layout problem: An integer linear program and Borda-Kendall-based method*. In: International Journal of Business and Systems Research 11/1-2(1-2):62–81

[LCW17] Li, J.; Chen, G.; Wu, Z.; He, X. (2017): *Distributed subgradient method for multi-agent optimization with quantized communication*. In: Mathematical Methods in the Applied Sciences 40/4(4):1201–1213

[Mal89] Malakooti, B. (1989): *Multiple objective facility layout: A heuristic to generate efficient alternatives*. In: International Journal of Production Research 27/7(7):1225–1238

[Mat15] Matai, R. (2015): *Solving multi objective facility layout problem by modified simulated annealing*. In: Applied Mathematics and Computation

[MSM13] Matai, R.; Singh, S.P.; Mittal, M.L. (2013): *A non-greedy systematic neighbourhood search heuristic for solving facility layout problem*. In: International Journal of Advanced Manufacturing Technology 68/5-8(5-8):1665–1675

[MG07] Meller, R.D.; Gau, K.-Y. (2007): *Facility layout objective functions and robust layouts*. In: International Journal of Production Research 34/10(10):2727–2742

[MJS16] Morrison, D.R.; Jacobson S.H.; Sauppe, J.J.; Sewell E.C. (2016): *Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning*. In: Discrete Optimization 19:79–102

[NO09] Nedić, A.; Ozdaglar, A. (2009): *Distributed subgradient methods for multi-agent optimization*. In: IEEE Transactions on Automatic Control 54/1(1):48–61

[OWM08] Ogryczak, W.; Wierzbicki, A.; Milewski, M. (2008): *A multi-criteria approach to fair and efficient bandwidth allocation*. Omega 36/3(3):451–463

[SHO15] Shchekutin, N.; Heinke, A.; Overmeyer, L. (2015): *Data transmission for a small-scaled, cyber-physical material handling system.* In: Proceedings on Symposium on Automated Systems and Technologies. Peter the Great St. Petersburg University

[PCP03] Pierreval, H.; Caux, C.; Paris, J.L.; Viguier, F. (2003): *Evolutionary approaches to the design and organization of manufacturing systems*. Computers and Industrial Engineering 44/3(3):339–364

[SZO16] Shchekutin, N.; Zobnin, S.; Overmeyer, L.; Shkodyrev, V. (2016): *Mathematical methods for the configuration of transportation systems with a focus on* *continuous and modular matrix conveyors*. In: Logistics Journal

[SS06] Singh, S.P.; Sharma, R.R.K. (2006): *A review of different approaches to the facility layout problems*. In: International journal of advanced manufacturing technology, 30/5-6(5-6):425–433

[SS10] Singh, S.P.; Singh, V.K. (2010): *An improved heuristic approach for multi-objective facility layout problem*. In: International Journal of Production Research 48/3/4(3/4):1171–1194

[SHE17] Sohrt, S.; Heinke, A.; Shchekutin, N.; Eilert, B.; Overmeyer, L.; Krühn, T. (2017): *Kleinskalige, cyber-physische Fördertechnik*. In: Handbuch Industrie 4.0. Springer Vieweg, Berlin, 21–44

[SVS05] Solimanpur, M.; Vrat, P.; Shankar, R. (2005): *An ant algorithm for the single row layout problem in flexible manufacturing systems*. In: Computers & Operations Research 32/3(3):583–598

[UTF10] Uriarte, C.; Thamer, H.; Freitag, M. (2010): *Fördertechnik aus der Zelle: Hochflexibles Fördersystem aus kommunizierenden und kooperierenden Modulen*

[UTF16] Uriarte, C.; Thamer, H.; Freitag, M., Thoben, K.-D. (2016): *Flexible Automatisierung logistischer Prozesse durch modulare Roboter*. Wissenschaftliche Gesellschaft für Technische Logistik

M. Sc. Nikita Shchekutin, Research Assistant at the Institute of Transport and Automation Technology, Leibniz Universität Hannover.
Nikita Shchekutin was born 1989 in Togliatty, Russia. Between 2006 and 2012 he studied Information technologies at the Peter the Great St. Petersburg Polytechnic University.

Address: Leibniz University Hannover, Institute of Transport and Automation Technology, An der Universität 2, 30823 Garbsen, Germany
Phone: +49 511 762-18339, Fax: +49 511 762-4007,
E-Mail: Nikita.shchekutin@ita.uni-hannover.de

M. Sc. Simon Sohrt, Research Assistant at the Institute of Transport and Automation Technology of Leibniz University Hannover.
Simon Sohrt was born 1987 in Neustadt am Rübenberge, Germany. Between 2011 and 2013, he studied mechanical engineering at Leibniz University Hannover.

Address: Leibniz University Hannover, Institute of Transport and Automation Technology, An der Universität 2, 30823 Garbsen, Germany
Phone: +49 511 762-18177, Fax: +49 511 762-4007,
E-Mail: simon.sohrt@ita.uni-hannover.de

Prof. Dr.-Ing. Ludger Overmeyer, Head of the Institute of Transport and Automation Technology of Leibniz University Hannover.
Ludger Overmeyer was born 1964 in Recke, Germany. Between 1984 and 1991, he studied electrical engineering at the University of Hannover. In 1996 he finished his doctorate in mechanical engineering at the University of Hannover. From 1997 to 2001 he worked as project manager, division manager and head of research and development at Mühlbauer AG in Roding. Since 2001 Ludger Overmeyer is Professor at the Institute of Transport and Automation Technology of Leibniz University Hannover.

Address: Leibniz University Hannover, Institute of Transport and Automation Technology, An der Universität 2, 30823 Garbsen, Germany
Phone: +49 511 762-3524, Fax: +49 511 762-4007,
E-Mail: ludger.overmeyer@ita.uni-hannover.de

DOI: 10.2195/lj_Proc_shchekutin_en_201710_01
URN: urn:nbn:de:0009-14-45987