

A comparison of evolutionary and simulated annealing algorithms for bi-criteria location-scheduling problem

Mirosław Ławrynowicz

Wroclaw University of Science and Technology
27 Wyb. Wyspińskiego St, 50-370 Wroclaw, Poland
Email: miroslaw.lawrynowicz@pwr.edu.pl

Grzegorz Filcek

Wroclaw University of Science and Technology
27 Wyb. Wyspińskiego St, 50-370 Wroclaw, Poland
Email: grzegorz.filcek@pwr.edu.pl

Abstract—A comparison of two heuristic algorithms solving a bi-criteria joint location and scheduling (ScheLoc) problem is considered. In this strongly NP-hard problem the sum of job completion times and location investment costs are used to evaluate the solution. The first solution algorithm (EV) uses an evolutionary approach, and the second more time-efficient algorithm (SA) is based on Simulated Annealing.

I. INTRODUCTION

IN RECENT years, the location-scheduling problem, referred to as ScheLoc, and its applications have attracted attention of many researchers (see e.g. [3], [5], [6], [10], [15], [20], [22]). The ScheLoc has been considered for the first time in [5]. Then it has been discussed and developed in many works. They differ depending on a kind of location area, type and number of machines, criterion evaluating a schedule of jobs, as well as used solution algorithms. A majority of works deals with a discrete area for the deployment of machines where a finite set of available positions for machines is known and given a priori, and a non-empty subset of this set is to be selected, (e.g., [6], [17], [15]). The evaluation of job schedule, noticed in the literature, is based on the makespan C_{max} and the sum of completion times $\sum C_j$ (see [21]). The criteria serve in the analyzed works as the assessment of ScheLoc as a whole. Some works include also other criteria for evaluation of the deployment of machines, (e.g., [11], [15]). Let us remind that the job scheduling sub-problem with different release dates is strongly NP-hard for a single machine and criterion $\sum C_j$ [16]. The other sub-problem is also NP-hard when the deployment of machines is treated as a particular case of the uncapacitated facility location problem (UFLP) or p-median problem (e.g., [13]). In consequence, ScheLoc which joins and extends those problems is at least as hard and complex as each of them, thus ScheLoc is strongly NP-hard. This fact justifies the development of efficient heuristic algorithms used to solve the problem. The ScheLoc problem has been also extended to multiple criteria versions, e.g. in [17] the expected value of $\sum C_j$ together with the total location costs of machines is considered, and in [25], four criteria are taken into consideration and solution algorithm, based on the NSGA II approach, has been developed.

This paper deals with one of the bi-criteria version of the ScheLoc problem. In this problem, it is assumed that a finite number of jobs are deployed at given original locations in a planar area. Every job has to be moved from its original location to the position of the corresponding machine site which is not known in advance. Then all the jobs moved to the same machine are scheduled. In this problem the number of machines is not given, but results from its solution. The machines can be deployed only in the locations from a given finite set, and not more than one machine can be launched at each location. The solution is evaluated by two criteria: the sum of jobs completion times and investment costs of deployment of machines in particular locations. To solve this strongly NP-hard problem (authors provide the proof in other work), two heuristic solution algorithms have been adopted and compared. The first algorithm (EV) is based on the evolutionary ([2]) approach, and the second (SA) on Simulated Annealing ([12]) approach. The algorithms are compared with the use of hyper-volume indicator (see [14]) to evaluate the quality of Pareto fronts obtained by the algorithms.

The remainder of the paper is organized as follows. The mathematical model is provided in Section II, which is followed by the presentation in Section III of the solution algorithms. Section IV is devoted to the computational experiments, which allowed us to evaluate the algorithms. Final remarks complete the paper.

II. PROBLEM FORMULATION

We consider a set $J = \{1, 2, \dots, j, \dots, n\}$ of n jobs, a set $A = \{a_1, a_2, \dots, a_j, \dots, a_n\}$ of their origin locations where $a_j = [a_j^{(1)}, a_j^{(2)}]^T$ represents the location of job j . The job j is characterized by execution time p_j , ready time ρ_j , and transportation speed v_j . The job needs to be performed by a single machine selected from a set of identical machines. The machines' prospective locations should be selected from a set $B = \{b_1, b_2, \dots, b_i, \dots, b_\mu\}$ of μ possible locations where $b_i = [b_i^{(1)}, b_i^{(2)}]^T$. It is assumed that the number m , $1 \leq m \leq \mu$ of employed machines is not a priori known.

Let us introduce a binary vector $y = [y_i]_{i=1, \dots, \mu}^T$ where $y_i = 1(0)$ if the location b_i is selected for the deployment of a

machine taken from the set of identical machines (otherwise). Consequently, the index i denotes a machine deployed at the location b_i . Let the schedule of jobs be represented by a three-dimensional binary matrix $x = [x_{jik}]_{j,k=1,n}^{i=1,\mu}$ in which current entry x_{jik} is equal 1(0) if the j th job is scheduled on the i th machine as the k th (otherwise). A performance of job j by machine i follows its transportation at a distance $d(a_j, b_i)$ between locations a_j and b_i if $y_i = 1$. The transport needs time $r_j(x, y) = \rho_j + \frac{1}{v_j} \sum_{k=1}^n \sum_{i=1}^{\mu} d(a_j, b_i) y_i x_{jik}$, which is interpreted as a release date for the job j . The p_j is the execution time of the job on the machine.

The evaluation of decisions x and y is done with the use of two criteria. The first one is the sum of completion times of all the jobs

$$q^{(1)}(x, y) \triangleq \sum_{i=1}^{\mu} \sum_{k=1}^n y_i C_{ik}(x, y) \quad (1)$$

where the auxiliary variable $C_{ik}(x, y)$ stands for the completion time of a job performed by the i th machine as the k th and depends on decision variables via constraints (5) and (6). The second criteria used is the total cost of all locations used by machines

$$q^{(2)}(y) \triangleq \sum_{i=1}^{\mu} c_i y_i \quad (2)$$

where c_i is the location cost of b_i .

The decision variables x and y must satisfy the following constraints:

$$\sum_{i=1}^{\mu} \sum_{k=1}^n x_{jik} = 1, j = 1, 2, \dots, n, \quad (3)$$

$$\sum_{j=1}^n x_{jik} \leq y_i, i = 1, 2, \dots, \mu, k = 1, 2, \dots, n, \quad (4)$$

$$C_{ik}(x, y) \geq \sum_{j=1}^n (r_j(x, y) + p_j) x_{jik}, \quad (5)$$

$$i = 1, 2, \dots, \mu, k = 1, 2, \dots, n,$$

$$C_{ik}(x, y) \geq C_{i,k-1}(x, y) + \sum_{j=1}^n p_j x_{jik}, \quad (6)$$

$$i = 1, 2, \dots, \mu, k = 2, 3, \dots, n,$$

$$1 \leq \sum_{i=1}^{\mu} y_i \leq \mu, \quad (7)$$

$$\sum_{j=1}^n (x_{j,i,k+1} - x_{jik}) \leq 0, \quad (8)$$

$$i = 1, 2, \dots, \mu, k = 1, 2, \dots, n-1,$$

$$y_i, x_{jik} \in \{0, 1\}, j, k = 1, 2, \dots, n, i = 1, 2, \dots, \mu. \quad (9)$$

Constraints (3) and (4) ensure that each job is performed on one position of a single machine in a launched location. The job j assigned to a single machine cannot start before its release date or completion time of the job scheduled to the same machine before j , what ensure constraints (5) and (6). The constraint (6) guarantees that the number of launched locations is between 1 and μ . To limit the number of equivalent solutions represented by matrix x , (8) is present. The last constraint defines the decision variables domains.

In consequence, the following bi-criteria optimization problem, referred to as BC_ScheLoc (Bi-Criteria ScheLoc), is

solved. Given $n, A, B, \mu, p_j, v_j, \rho_j, c_i, j = 1, 2, \dots, n, i = 1, 2, \dots, \mu$ find the schedule of jobs x and the locations of machines y minimizing a vector of criteria $q(x, y) = [q^{(1)}(x, y), q^{(2)}(y)]^T$ subject to constraints (3)-(9). Let us point out that the resulting from the optimisation number of used machines m can be calculated using the formula $m = \sum_{i=1}^{\mu} y_i$.

III. SOLUTION ALGORITHMS

Considered bi-criteria optimization problem, as it was stated in the Introduction, is strongly NP-hard. Hence, the general schemes of an evolutionary approach [2] and the Simulated Annealing metaheuristic [12] with some improvements are used.

A. Algorithm EV

The proposed algorithm uses a general evolutionary approach. The next subsection provide information about encoding of the chromosome, and used selection, mutation, and crossover operators, as well as stop condition.

1) *Encoding*: A chromosome $E = (e_1^m, \dots, e_{\mu}^m, e_{\mu+1}^s, \dots, e_{\mu+n}^s, e_{\mu+n+1}^{job}, \dots, e_{\mu+2n}^{job}) \triangleq ((e^m), (e^s), (e^{job}))$ encoded as a three-part sequence represents a candidate solution (y, x) . It contains binary values e^m and e^s along with integer values from the set J for e^{job} . The binary values e_i^m directly represent y_i and allow calculating m . The value of $\sum_{l=1}^t e_l^m = i$ calculated for every $e_t^m = 1$ indicates the index i of the machine deployed at location t . The following mappings decode the optimization variable y and the number of launched locations (machines) m :

$$y = f_y(E) = [e_i^m]_{i=1,\mu}^T, \quad (10)$$

$$m = f_m(E) = \sum_{t=1}^{\mu} e_t^m. \quad (11)$$

The $m-1$ '1's in e^s located at positions $w_i, i = 1, 2, \dots, m-1$ and referred to as indices of separation specify the division of the set of all jobs into m subsequences of jobs assigned to individual machines. Namely, the i th index of separation indicates the first position of the $(i+1)$ th subsequence of e^{job} representing jobs assigned to the machine deployed at the opened location pointed out by the $(i+1)$ th in order '1' in e^s . The first subsequence of jobs in e^{job} , assigned to the machine pointed out by the first in order '1' in e^m , starts at $e_{\mu+n+1}^{job}$. Consequently, the three-dimensional binary matrix x can be retrieved from E by the mapping:

$$x = f_x(E) = \left[x_{j,i+1,k} = \begin{cases} 1, & \text{if } \begin{aligned} j &= e_{\mu+n+w_i+k-1}^{job} \\ k &= 1, 2, \dots, w_{i+1} - w_i, \\ w_0 &= 1, \\ i &= 0, 1, \dots, m-1, \end{aligned} \\ 0, & \text{otherwise.} \end{cases} \right] \quad (12)$$

The algorithm searches solutions iteratively and independently for each subproblem that takes into account the fixed number of machines. It starts with the generation of an

initial population $\tilde{S}_i = \left\{ \tilde{E}_{il} \right\}_{l=1, \dots, \alpha_i}$ of size α_i , $i = 1, 2, \dots, \mu$, which cardinality $|\tilde{S}_i| = \alpha_i$ is equal for each subproblem. The creation of solutions is made by uniformly setting the values within the genome with the assumption about the Hamming distance between each pair of chromosomes higher than ϑ . This diversification strategy helps to avoid premature convergence during the initial iterations.

2) *Evolutionary operators*: The Stochastic Universal Sampling method for equally weighted criteria is employed for the selection process [1]. Next, the chromosomes undergo a parallel crossover with the use of two different operators with the probability $\varphi_c = 0.95$ [4]. For the binary parts, the Count-Preserving Crossover (CPC-2) operator is applied [8]. For the integer part, the Order-based Crossover Operator (OX2) recombines solutions [23]. The Simple Inversion Mutation operator (SIM) changes the integer part of E with probability $\varphi_m = 0.01$ [7]. The binary parts of E are randomly modified by the Swap Mutation (SM) operator [19].

Finally, the stop condition is fulfilled if there is no improvement of solutions representing the Pareto front through Γ consecutive iterations. The parameter Γ_{max} restricts the maximal number of iterations for each subproblem.

B. Algorithm SA

The generation of the initial solutions set I involves a random assignment of all the jobs to each possible number of machines $1, 2, \dots, \mu$. For $s \in \{1, 2, \dots, \mu\}$, the different initial solutions are created K times, $|I| = K\mu$. The time of performing computations is constrained by r^{max} and the cooling schedule $T(i) = T(0) - \gamma i$ where i stands for the current iteration index, γ is the adjustable parameter and, the initial temperature $T(0)$ chosen experimentally so that an acceptance probability of worse solutions is close to one in the first iteration [9], [12]. The set $S(i)$ stores each best found (non-dominated in Pareto's sense) pair (x, y) of the decision variables. The new solution (\tilde{x}, \tilde{y}) referred to as a neighboring solution of (x, y) is determined by reassignment of all the jobs being processed on the β (adjustable parameter) machines providing the largest sum of completion times:

$$\begin{aligned} & \sum_{k=1}^n y_{s_w} C_{s_w k}(x, y) \geq \sum_{k=1}^n y_{s_{w-1}} C_{s_{w-1} k}(x, y) \geq \dots \\ & \geq \sum_{k=1}^n y_{s_1} C_{s_1 k}(x, y), \beta < w, \forall l \in \{s_w, s_{w-1}, \dots, s_1\} (y_l = 1), \end{aligned} \quad (13)$$

to the nearest β unoccupied feasible locations. The machines at new locations $\tilde{y}_{s_k}, \tilde{y}_{s_{k-1}}, \dots, \tilde{y}_{s_\beta}$ receive the jobs from the set $\tilde{J} = \{j \in J : x_{jlk} = 1, k = 1, 2, \dots, n, l = s_w, s_{w-1}, \dots, s_\beta\}$. The reassignment preserves a greedy approach and requires $|\tilde{J}|$ steps. At each step, the job $j \in \tilde{J}$ is assigned in such a way that (2) is minimized with the exclusion of non-chosen jobs from \tilde{J} . The neighboring solution $(\tilde{x}, \tilde{y}) \in N_{(x,y)}$ replaces $(x, y) \in S(i)$ only if the dominance condition is met:

$$E(x, y, \tilde{x}, \tilde{y}) = \begin{cases} true, & q^{(1)}(\tilde{x}, \tilde{y}) < q^{(1)}(x, y) \\ & \vee q^{(2)}(\tilde{y}) < q^{(2)}(y) \\ false, & otherwise \end{cases} \quad (14)$$

Consequently, the decisions (\tilde{x}, \tilde{y}) may undergo further changes only if the acceptance probability $(1 + \exp(\frac{Dist(\tilde{x}, \tilde{y}) - Dist(x, y)}{T(i)}))^{-1}$ exceeds a value of the acceptance threshold Λ and $Dist(x, y)$ means the Euclidean distance from the ideal point $(0, 0)$. The introduced probability function is the modified version of the function described in [18] that enables an evaluation of a bi-criteria solution.

Finally, the iteration count depends directly on the initial temperature $T(0)$. Moreover, the calculation time is constrained by the parameter r^{max} .

C. Tuning procedure

The tuning procedure has been conducted for each dataset, which is based on the offline approach described in detail in [24]. Three parameters of the EV with restricted domains of their possible values have been selected for tuning: $\alpha_i \in \{100, 150, \dots, \mathbf{350}, \dots, 500\}$, $\tilde{\alpha} = \{0.6\alpha_i, 0.7\alpha_i, \mathbf{0.8}\alpha_i, 0.9\alpha_i\}$, $\Gamma \in \{20, 30, \dots, \mathbf{50}, \dots, 80\}$, where the default values are marked by bold types. The default values of other parameters have been assumed as follows: $\varphi_c = 0.95$, $\varphi_m = 0.01$, $\Gamma_{max} = 3500$, $\vartheta = 0.1(n + \mu)$. Analogously, three parameters of the SA have been selected for tuning: $\gamma \in \{0.85, \mathbf{0.9}, 0.95, 0.99\}$, $T(0) \in \{10^2, 10^3, \mathbf{10^4}, 10^5\}$, $\beta \in \{30, \mathbf{50}, \dots, 100\}$. The default values of other parameters have been assumed as follows: $r^{max} = 15$ min, $K = 20$, $\Lambda = 0.5$.

IV. COMPUTATIONAL EXPERIMENTS

The purpose of the conducted computational experiments is the comparison of the EV and SA. Research includes the detailed analysis of Pareto fronts generated by the developed algorithms. The definition of quality indices and the instance generation assumptions in Subsection IV-A is followed by the evaluation in Subsection IV-B. All the computations have been made using a PC with AMD Ryzen Threadripper 2970WX equipped with 32GB of RAM. The EV and SA have been implemented in the Haskell language.

A. Foundations of computational experiments

The parameters values have been randomly generated according to the uniform distribution. The locations of jobs a_j , and available locations for machines b_i , have been drawn from set $[0, 1000]$, job processing times p_j from $\{20, 21, \dots, 50\}$, costs of machine locations c_i from $\{70, 71, \dots, 139\}$. The speed of job j movement can be calculated with the formula $v_j = 350/p_j$. We have also assumed the domains for the number of jobs n and the number of available locations for machines μ as $\{25, 50, \dots, 150\}$ and $\{5, 10, \dots, 25\}$, respectively. Finally, $\rho_j = 0$ for all the jobs.

Let us introduce a Pareto front $PF \triangleq (q_l = (q_l^{(1)}, q_l^{(2)})) : q_{l-1}^{(1)} < q_l^{(1)} < q_l^{(2)} < q_{l+1}^{(2)}$, which is an L -element sequence of points, resulted from the run of solution algorithm. We distinguish within PF the point q_λ , which is the closest to $(0, 0)$ according to the Euclidean distance $Dist = \sqrt{(q_\lambda^{(1)})^2 + (q_\lambda^{(2)})^2}$. The values of q_λ express the trade-off between (1) and (2). For evaluation we propose a well-known hypervolume indicator

TABLE I
DEPENDENCE OF $I_{H,EV}$ AND $I_{H,SA}$ ON μ AND n

(μ, n)	$I_{H,EV}$			$I_{H,SA}$			$1 - (I_{H,SA}/I_{H,EV})$
	Max	Min	Avg	Max	Min	Avg	Avg
(5,25)	0.794	0.703	0.763	0.760	0.686	0.733	0.039
(5,50)	0.779	0.711	0.756	0.757	0.702	0.740	0.021
(5,75)	0.721	0.654	0.702	0.706	0.621	0.656	0.066
(5,100)	0.736	0.661	0.711	0.740	0.678	0.714	-0.004
(5,125)	0.701	0.655	0.682	0.718	0.681	0.701	-0.028
(5,150)	0.727	0.659	0.703	0.712	0.643	0.687	0.023
(15,25)	0.807	0.745	0.780	0.778	0.702	0.734	0.059
(15,50)	0.769	0.699	0.739	0.761	0.691	0.736	0.004
(15,75)	0.708	0.645	0.679	0.712	0.649	0.688	-0.013
(15,100)	0.687	0.622	0.667	0.697	0.622	0.681	-0.021
(15,125)	0.717	0.636	0.683	0.699	0.629	0.659	0.035
(15,150)	0.735	0.659	0.704	0.707	0.621	0.673	0.044
(25,25)	0.691	0.636	0.655	0.667	0.612	0.632	0.035
(25,50)	0.713	0.649	0.673	0.689	0.617	0.635	0.056
(25,75)	0.699	0.624	0.651	0.669	0.587	0.641	0.015
(25,100)	0.689	0.603	0.639	0.699	0.609	0.646	-0.011
(25,125)	0.667	0.599	0.634	0.670	0.603	0.639	-0.008
(25,150)	0.706	0.631	0.671	0.688	0.617	0.649	0.033

TABLE II
DEPENDENCE OF $Dist_{EV}$ AND $Dist_{SA}$ ON μ AND n

(μ, n)	$Dist_{EV}$			$Dist_{SA}$			$(1 - Dist_{SA}/Dist_{EV}) * 100\%$
	Max	Min	Avg	Max	Min	Avg	Avg
(25,25)	951	934	945	957	927	945	0.01
(15,25)	952	934	940	944	918	933	0.71
(5,25)	985	944	970	981	952	965	0.48
(25,50)	1959	1915	1934	1971	1896	1936	-0.11
(15,50)	1931	1896	1916	1929	1875	1905	0.56
(5,50)	2886	2824	2858	2942	2791	2867	-0.34
(25,75)	2986	2909	2942	3018	2920	2971	-0.98
(15,75)	3089	2991	3043	3037	2980	3017	0.84
(5,75)	6191	5995	6074	6314	5927	6131	-0.94
(25,100)	3978	3929	3954	4035	3882	3961	-0.18
(15,100)	4866	4753	4821	4730	4530	4662	3.29
(5,100)	10810	10312	10500	10915	10241	10508	-0.08
(25,125)	5388	5149	5251	5365	5146	5278	-0.53
(15,125)	7076	6987	7046	7077	6686	6927	1.68
(5,125)	16352	16021	16135	16465	15686	16102	0.20
(25,150)	7042	6977	7011	7090	6631	6903	1.53
(15,150)	9976	9859	9902	9984	9396	9635	-0.53
(5,150)	23477	22844	23059	23781	22205	22897	0.70

I_H [14] and $Dist$ as quality indices. Indicator I_H uses normalized values of criteria and measure the part of the criteria area weakly dominated by an evaluated Pareto front PF. Its analytic formula is $I_H = \sum_{l \in \overline{1,L}} (1 - \bar{q}_l^{(1)}) (\bar{q}_l^{(2)} - \bar{q}_l^{(1)})$

where $\bar{q}_0^{(2)} = 1$, $\bar{q}_l^{(i)} = q_l^{(i)} / q_{\max}^{(i)}$, $i = 1, 2$ are normalized values of both criteria for PF. Normalization is done with $q_{\max}^{(2)} = \sum_{i=1}^{\mu} c_i$, and $q_{\max}^{(1)} = \max_{i \in \{1,2,\dots,\mu\}} \left\{ \max_{j \in \{1,2,\dots,n\}} n * (\rho_j + \frac{1}{v_j} \sum_{k=1}^n \sum_{i=1}^{\mu} d(a_j, b_i)) \right\} + \sum_{j=1}^n (n + 1 - j) * p_{s_j}$, where the processing times $p_{s_1} \geq p_{s_2} \geq \dots \geq p_{s_n}$ are sorted in descending order. Increasing value of the I_H arranges a greater area of dominated solutions and an evenly Pareto front. In effect, the I_H is desired to be close to 1. Smaller $Dist$ values are preferred as the trade-off point is then closer to the reference point $(0, 0)$.

B. Results of experiments

The evaluation of the EV and SA comprises the comparison of quality indices calculated for Pareto front PF_{EV} generated by the EV with the corresponding indices calculated for Pareto front PF_{SA} generated by the SA. Additionally, we apply processing time $Time$ as the performance indicator. The indices calculated for the EV and SA are marked by the subscripts EV and SA. The values of both criteria and quality indices are presented in Tables I-II, where the Avg, Min, and Max mean average, minimum, and maximum, respectively. Each value is obtained through 30 independent runs of the EV and SA. Table I presents the Avg, Min, and Max values of the indicator I_H for the EV and SA, and comparison between them in average as $1 - (I_{H,SA}/I_{H,EV})$. With reference to this indicator, both algorithms have achieved similar results as values in the last column are very small (the difference

TABLE III
DEPENDENCE OF COMPUTATIONAL TIMES (IN MILLISECONDS) T_{EV} AND T_{SA} ON μ AND n

(μ, n)	T_{EV}			T_{SA}			T_{EV}/T_{SA}
	Min	Max	Avg	Min	Max	Avg	Avg
(5,50)	8246	10281	9124	6016	6804	6287	1.45
(5,150)	27526	32046	30822	20397	22219	21016	1.47
(15,50)	36251	40403	38900	17005	17664	17211	2.26
(15,150)	96155	102302	98669	44753	45996	45256	2.18
(25,50)	75371	104258	85577	28089	30676	29232	2.93
(25,150)	159050	181497	172228	57998	59936	59268	2.91

between algorithms does not exceed 6,6% in favour of the EV). In majority of analyzed instances, the EV algorithm outperforms SA, but it cannot be concluded that EV is better. For some instances, the SA gets better results for all the indicators. The analysis of Table II shows that both algorithms have reached similar results and there is no possibility to indicate the better approach. However, the SA has ensured better results in most cases. The slight differences between the both algorithms do not exceed 3.3% on average. In addition, the comparison of computation times illustrated in Table III shows that the SA has surpassed the EV. In some cases, the EV has been about 3 times slower than SA, and the difference increases along with the size of the problem instance.

V. FINAL REMARKS

The primary contribution of this study deals with the analysis of evolutionary and metaheuristic-based approaches for the bi-criteria Scheloc problem. Two algorithms have been developed and evaluated with the use of the hypervolume I_H , $Dist$ and $Time$ indicators. As the SA has ensured faster execution times, it is strongly recommended for the critical-time applications. In addition, the minimal distance from the highly preferable point (0,0) has been mainly delivered by the SA. On the contrary, the evenly arranged Pareto fronts have been provided by the EV. Although the unambiguous interpretation of execution times, the better algorithm cannot be clearly chosen on the basis of the I_H and $Dist$ indicators. Finally, further research may include other criteria or seeking for more efficient approximation schemes.

REFERENCES

- [1] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*, pages 14–21, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.
- [2] K. Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001.
- [3] D. Elvikis, H. W. Hamacher, and M. T. Kalsch. Simultaneous scheduling and location (ScheLoc): the planar ScheLoc makespan problem. *Journal of Scheduling*, 12(4):361–374, 2009.
- [4] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, Jan 1986.
- [5] H. Hennes and H. Hamacher. *Integrated Scheduling and Location Models: Single Machine Makespan Problems*. Report in Wirtschaftsmathematik. Univ., Fachbereich Mathematik, 2002.
- [6] C. Heßler and K. Deghdak. Discrete parallel machine makespan ScheLoc problem. *Journal of Combinatorial Optimization*, 34(4):1159–1186, 2017.
- [7] J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, 1975.
- [8] Y.-C. Hou and Y.-H. Chang. A new efficient encoding mode of genetic algorithms for the generalized plant allocation problem. *Journal of Information Science and Engineering*, 20:1019–1034, 09 2004.
- [9] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations research*, 37(6):865–892, 1989.
- [10] M. T. Kalsch. *Scheduling-Location (ScheLoc): Models, Theory and Algorithms*. Verlag Dr. Hut, 2009.
- [11] M. T. Kalsch and Z. Drezner. Solving scheduling and location problems in the plane simultaneously. *Computers & Operations Research*, 37(2):256–264, 2010.
- [12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [13] J. Krarup and P. M. Pruzan. The simple plant location problem: Survey and synthesis. *European Journal of Operational Research*, 12(1):36 – 81, 1983.
- [14] M. Laszczyk and P. B. Myszowski. Survey of quality measures for multi-objective optimization. Construction of complementary set of multi-objective quality measures. *Swarm and Evolutionary Computation*, 48:109–133, 2019.
- [15] M. Ławrynowicz and J. Józefczyk. A memetic algorithm for the discrete scheduling-location problem with unrelated executors. In *Proc. of 24th Int. Conf. on Models and Methods in Automation and Robotics MMAR*, pages 158–163, 2019.
- [16] J. K. Lenstra, A. R. Kan, and P. Brucker. Complexity of machine scheduling problems. In *Annals of discrete mathematics*, volume 1, pages 343–362. Elsevier, 1977.
- [17] M. Liu, X. Liu, E. Zhang, F. Chu, and C. Chu. Scenario-based heuristic to two-stage stochastic program for the parallel machine ScheLoc problem. *International Journal of Production Research*, 57(6):1706–1723, 2019.
- [18] Z. Michalewicz and D. B. Fogel. *How to solve it: modern heuristics*. Springer Science & Business Media, 2013.
- [19] I. M. Oliver, D. Smith, and J. R. Holland. Study of permutation crossover operators on the traveling salesman problem. In *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms at the Massachusetts Institute of Technology, Cambridge, MA. Hillsdale, NJ: L. Erlbaum Associates, 1987*.
- [20] B. Piasecki and J. Józefczyk. Evolutionary algorithm for joint task scheduling and deployment of executors. In: *Automation of Discrete Processes. Theory and Applications*, Silesian University of Technology, 1:169–178, 2018.
- [21] M. Pinedo. *Scheduling: theory, algorithms and systems development*, volume 29. Springer-Verlag NY, 2012.
- [22] M. Rajabzadeh, M. Ziaee, and A. Bozorgi-Amiri. Integrated approach in solving parallel machine scheduling and location (ScheLoc) problem. *International Journal of Industrial Engineering Computations*, 7(4):573–584, 2016.
- [23] G. Syswerda. Scheduling optimization using genetic algorithms. *Handbook of genetic algorithms*, pages 332 – 349, 1991.
- [24] E.-G. Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.
- [25] S. Wesolowski, N. Francetić, and S. C. Grant. TraDE: Training device selection via multi-objective optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 2617–2624. IEEE, 2014.