Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

A NEW METHODOLOGY FOR APPLYING SIMULATION DRIVEN METAHEURISTICS TO THE BALANCING OF SECURITY INSPECTION LINES

Stefano Saetta
Lorenzo Tiacci

Dipartimento di Ingegneria Industriale
Università di Perugia
Via Duranti 67
06125 Perugia, ITALY

ABSTRACT

In the paper author show a new approach to the line balancing of security inspection lines. In this approach there is a particular combination of simulation of metaheuristics and modeling and simulation. Simulation is used directly during the metaheuristics process for evaluating the objective function. In this manner it is possible to obtain more effective objective function This process can be quite fast in many situations where simulation calculation can be done very effectively. In particular simulation is used for the computation of the objective function. In particular in the paper a tabu search based algorithm is considered. The application concern a problem of line balancing. Same algorithm is applied to a conventional objective function in order to show benefits of the new approach.

1.0 INTRODUCTION

Studies concerning security operations are becoming the more and more important especially in the transportation sector because of the necessity to guarantee a fast and simple traveling system while reaching the highest security level. One of must time consuming operation are inspections. Inspections times can have an high variability depending on the product being checked. Considering, for instance a baggage check procedure. The inspection procedure will depend on the type of the baggage and also on the things inside the baggage. In order to face such a kind of problems algorithms used for the line balancing problems can be used. They can include aspects such as the paralleling or the mixed products situation that are very similar to the baggage security screening problems. The efficiency of the baggage security screening, for instance, is a focal point in the security process in airports (Leone and Liu 2005). In order to face balancing complex problems metaheuristics such simulated annealing, tabu search, genetic algorithm, etc. have been used. It is possible to increase the performance of such algorithm by using simulation to generate results to be used in the objective function. This results can be more reliable than conventional performance indicators achieved without simulation.

In the paper a simple tabu search model for calculating the value of the objective function for a problem of balancing in a security inspection lines (such as, for instance, a checked baggage security screening service) is developed. Methods show could be applied with others methods and others security problems.

The Security Inspections Operations (SIO) are very similar as problem to the Assembly line balancing (ALB) problem. In ALB as in SIO is the problem of assigning various tasks to work-centers (or stations), while optimizing one or more objective functions without violating any restrictions imposed on the line. LI speed of the security line and length of each uniform station dictates time available per operator, known as cycle time. A first classification of LI problem can be deducted from (Sholl and Becker 2003). It can be made according to two main different objectives. Therefore, we have:

type 1 problems: to minimize the amount of workers required on the security line, given a specific cycle time.

type 2 problems: to minimize the cycle time, given a specified number of workers.

type G problems: a more general type is obtained by minimizing the sum of idle times subject to varying inspections rates and numbers of stations.

The simple LI problem has been modified into more complex problems GLI (generalized line inspection) as can be argued from (Becker and Sholl 2003). In the following there is a description of the most important modeling options.

1.1 Paralleling

Paralleling is the possibility of allowing multiple workers to be assigned to a single station (i.e. to double the work capability of a station or to have two identical parallel stations). The use of paralleling gives more flexibility to the problem and allows the existence of tasks with an execution time greater than cycle time, but it increases the problem complexity.

1.2 Mixed model/Multi model

If several product types are inspected on the same line then the problem is called mixed- or multi-model. The difference between these two problems is in the security sequence that is used. A mixed model line uses IS random sequence, i.e. the units of different models are sequenced in an arbitrarily way, while in multi-model line the units of the same type are checked in batches with intermediate setup operations.

1.3 Incompletion handling

Incomplete tasks can generate two main situation. In the first case, if some tasks are not completed, no blocking occurs. The completion is realized off the line with a certain cost. Otherwise cycle time is extended, and a blocking of the previous and following stations occurs.

1.4 Stochastic task times

It is usual to consider the task times as stochastic variables. The most frequently assumed distribution is normal distribution. It is also usual to give the standard deviation σ as a fraction of the mean time, that is:

$$\sigma = cv \cdot \mu$$

where μ is the mean task time and cv is the coefficient of variation. It often assumes a value between 0.1 and 0.35.

A lot of algorithms and methods for finding solution for the IL problem can be found in literature. Most of them find solutions for the GLI problem, but very few has faced the mixed-model problem with parallel stations and stochastic task times. For example: the heuristic method proposed by (Askin and Zhou 1997) that solves mixed-model problems with paralleling; the two-stage heuristic method proposed by (Vilarinho and Simaria 2002) that also includes zoning constraints; the algorithms proposed by McMullen, who solves the problem with a heuristic method (McMullen and Frazier 1997), with a simulated annealing approach (McMullen and Frazier 1998) and with ant techniques (McMullen and Tarasewich 2003).

In this paper we propose a new approach: using the growing computational potential of personal computers, it is possible to quickly run a simulation of the layout that represents the current solution at each step of the algorithm procedure. Results of simulation can be utilized to calculate the objective function that includes "dynamic" parameters, i.e. calculated using simulation outputs (throughput, actual utilization, flow time, etc.), rather than "static" parameters (design cost, smoothness index, etc.). We implemented this procedure with a tabu-search algorithm and used it to solve IL mixed-model problems with parallel stations and stochastic task times. Then, we compare the solutions obtained by the algorithm that uses the "dynamic" objective function with the one that uses the static objective function.

2. NOTATION

ct cycle time;
n number of tasks of the problem;
m number of models;
p number of stations;
t_{ij} time required by model j in station i;
α_j demand proportion for model j;

$t_{\bullet j} = \sum_{i=1}^{p} t_{ij}$ total time required by model j;

$t_{\bullet\bullet} = \sum_{j=1}^{m} \alpha_j t_{\bullet j}$ total weighed inspection time;

w_i number of workers in station i;

$w_\bullet = \sum_{i=1}^{p} w_i$ total number of workers in the line;

n_i number of tasks assigned to station i;

$e_i = w_i \cdot n_i$ number of equipments in station i;

$e_\bullet = \sum_{i=1}^{p} e_i$ total number of equipment in the line;

LC labor cost;
EC equipment cost;
$DC = LC \cdot w_\bullet + EC \cdot e_\bullet$ the design cost;
ts tabu size;
ls leash size;
TP actual throughput;
$Util$ actual line utilization;
$LB_w^{'}$ theorist lower bound for workers;
LB_w next integer to $LB_w^{'}$;
LB lower bound for design cost;
MV_p model variability;
$Util_s$ line utilization;

OF_s static objective function;

OF_d dynamic objective function.

3 METHODOLOGY

In this section we describe the algorithm, the simulation model and the outputs used to calculate the objective function.

3.1 The Algorithm

The proposed methodology is based on a classical tabu-search technique (Chiang 1998), applied to assembly line balancing problem. It uses a set of rules that allows an intelligent exploration of the solution space of a certain problem. Glover] presented it in its current form in 1989 (Glover 1989,a) (Glover 1989,b). It always starts from a random or a heuristic solution, and then it modifies that solution, doing little changes. The solution is modified in two ways: by the shift of a task from a station to another one, or by the change of two tasks of different stations. The main idea is to prevent, at each step of the algorithm, to search solution in pathways yet explored, in order to avoid to be trapped in local optima. This is possible by implementing a flexible memory in the algorithm that remembers the previous solution changes.

We implemented it with an integer matrix (called tabu-matrix) that has a number of lines equal to the number of problem tasks and a number of columns equal to the number of stations in the previous solution. The value of an element indicates if a task can be put in a certain station. For example, if the matrix has a value equal to 0 in the element corresponding to task k and station i, then task k can shift form the current station to the station i; on the contrary if the value is 12, then it will be able to shift in station i only after the 12th iteration. When a shift occurs, the corresponding value into the tabu-matrix (that was 0) become equal to the number of the current iteration plus a constant (called tabu-size). This means that this task can not return to the previous station before that a number of iterations equal to the tabu-size have been performed. It is very important to assume the appropriated value for the tabu-size. In effect, if it is too small, then the algorithm will be often trapped in local optima, while if it is too large, then it is possible to lose a lot of good solutions. This problem is partially solved using the aspiration criterion. This means that, if a random shift gives the best solution found until that iteration, then the solution is accepted even if the shift is forbidden. Moreover the tabu-size must be chosen in relation to the problem size. In fact, a good tabu-size for a 15 tasks problem is too small for a 75 tasks problem.

We remember that this algorithm solves GIL mixed-model problem with stochastic tasks times and parallel stations. In particular, the use of paralleling generates some problem because the space of solutions becomes very large and the algorithm loses its effectiveness; in effect the number of iterations is not unlimited, and the algorithm needs a criterion that prevents to search for too many iterations towards worst solutions. Therefore we implemented another new criterion that we called the "leash" criterion. Suppose that the algorithm is at iteration i. If the algorithm, from iteration i to iteration i+ls, with *ls* positive integer, finds solutions that do not improve the objective function, then it comes back to the current solution of iteration i. We called ls the leash-size. In this way we can join random search to objective function without binding the algorithm.

3.2 The Simulation Model

In order to calculate a dynamic objective function, a discrete event simulation model has been implemented. This model follows the process based approach, i.e. it is constituted by a set of process that represent the life cycle of the system entities. This type of approach is easy to implement with object-oriented program language, so we used the Java package javaSimulation (Helsgaun 2004). The model allows simulating an inspection line with no buffer between two different stations. Task times are considered stochastic, with $cv=0.3$. The line allows inspections various kinds of models (mixed model line). If a delay occurs in a station the inspection is not completed out of the line, but the precedent and the following stations wait until the completion of the tasks.

3.3 The Outputs and the Objective Function

The outputs of a single simulation are the throughput (*TP*), i.e. the mean time between two consecutive completed inspections, and the actual utilization (*Util*), i.e. the ratio between the time in wich an operator is busy and the total available working time. At each iteration the current solution is simulated and the mean value of *TP* and *Util* is calculated. The objective function OF_d is a weighted mean of the design cost and these two outputs. Let

$LB_w' = \frac{t_{\bullet\bullet}}{ct}$ the non integer lower bound for the number of workers and

LB_w the next integer to LB_w'.

Then

$$LB = LC \cdot LB_w + EC \cdot n$$

is a lower bound for the design cost.

The objective function to be maximized is:

$$OF_d = c_1 \cdot \frac{LB}{DC} + c_2 \cdot \frac{ct}{TP} + c_3 \cdot Util \quad (1)$$

where c_1, c_2, c_3 are three constants and $c_1 + c_2 + c_3 = 1$. In literature there are many examples that show how genetic algorithms often find better solutions for the IL problem.

4. DESIGN OF EXPERIMENT AND RESULTS

We tested the proposed algorithm on some classical problems taken from Sholl data set (see table 1.) (Sholl 1993). However, with respect to the original problem that was developed for a single model line with no paralleling, the data set has been modified in order to consider 4 different models and paralleling allowed.

In table 1 it is possible to see the values of tabu-size and leash-size for each problem.

Table 1 – Problem Data Set

Problem Name	n	ct	ts	ls	Number of models
Roszieg	25	18	4	50	4
Roszieg	25	14	4	50	4
Roszieg	25	11	4	50	4
Roszieg	25	9	4	50	4
Gunther	35	54	5	70	4
Gunther	35	41	5	70	4
Gunther	35	36	5	70	4
Gunther	35	30	5	70	4
Wee-mag	75	34	12	150	4
Wee-mag	75	28	12	150	4
Wee-mag	75	24	12	150	4
Wee-mag	75	21	12	150	4

We used the same algorithm with a "static" objective function in order to show that we get better solution using the "dynamic" objective function. It is important to underline that the algorithm uses in both cases similar CPU time-resources. The dynamic objective function is defined by (1). TP and $Util$ of the current solution are calculated via simulation. Each current solution is simulated once for 120 hours of inspection time. Data collection starts only when steady state is achieved. As in the algorithm, also in the simulation the task times are considered stochastic variables normally distributed with $cv = 0.3$. The static objective function we used is the following:

$$OF_s = c_1 \cdot \frac{LB}{DC} + c_2 \cdot \frac{LB_w}{MV_p} + c_3 \cdot Util_s \quad (2)$$

where:

$$MV_p = \frac{\sum_{j=1}^{m} \alpha_j \cdot \sum_{i=1}^{p} \left(\frac{t_{ij}}{w_i} - \frac{t_{\bullet j}}{w_\bullet} \right)^2}{ct^2}$$ is the model variability and

$$Util_s = \frac{LB_w}{w_\bullet}$$ is the static line utilization.

Model variability was introduced by Bukchin and is a measure of the smoothness in the line for each model, weighted by the model demand proportion. The formulation here proposed has been modified for solution with paralleling. In (Bukchin 1998) it is showed that the model variability is one of the indexes with the highest correlation with simulated TP in mixed model inspection line. In the same way static line utilization has the highest correlation with $Util$, so we used these two parameters in substitution of the simulation outputs. The results are showed in tables 2, 3 and 4.

Table 2 and 3 show respectively the final outputs of solutions found by algorithm that uses the static and the dynamic objective functions. The values are the mean of 100 iterations.

Table 4 shows the comparison between the values of the objective function simulated for both solutions, static and dynamic.

It should be evident that for each problem considered the algorithm with dynamic objective functions provides a better solution than the other one (remember that the function has to be maximized)

Table 2. Tabu-search with static objective function.

Problem Name	Iter.	Design Cost	MV	Static Util	TP	Util
Roszieg	2500	477	10,29	0,689	16,64	0,534
Roszieg	2500	573	12,16	0,655	13,97	0,525
Roszieg	2500	699	15,25	0,655	10,89	0,541
Roszieg	2500	810	18,29	0,669	9,13	0,534
Gunther	3500	627	10,62	0,658	48,83	0,526
Gunther	3500	879	12,08	0,706	35,86	0,512
Gunther	3500	846	17,05	0,666	34,61	0,497
Gunther	3500	1.107	17,20	0,654	26,02	0,474
Wee-mag	7500	2.718	46,66	0,614	38,80	0,604
Wee-mag	7500	3.417	45,70	0,626	30,21	0,564
Wee-mag	7500	3.648	59,68	0,619	27,19	0,589
Wee-mag	7500	4.017	66,99	0,632	24,80	0,611

Table 3**.** Tabu-search with dynamic objective function

Problem Name	**Iter.**	**Design Cost**	**MV**	**Static Util**	**TP**	**Util**
Roszieg	250	447	9,86	0,648	16,76	0,496
Roszieg	250	567	12,81	0,613	12,75	0,496
Roszieg	250	738	14,67	0,603	9,69	0,494
Roszieg	250	888	18,78	0,604	8,05	0,579
Gunther	350	615	11,51	0,599	46,45	0,512
Gunther	350	870	13,63	0,609	32,87	0,471
Gunther	350	921	15,60	0,588	31,37	0,463
Gunther	350	1.065	16,44	0,639	27,16	0,531
Wee-mag	750	2.730	48,56	0,476	36,22	0,551
Wee-mag	750	3.423	46,29	0,497	26,72	0,491
Wee-mag	750	3.834	60,57	0,453	24,43	0,568
Wee-mag	750	4.137	67,33	0,641	21,33	0,532

Table 4. Comparison between solutions

Problem Name	**Static O.F.**	**Dynamic O.F.**	**Difference**
Roszieg	0,724	0,726	0,002
Roszieg	0,699	0,722	0,023
Roszieg	0,716	0,726	0,010
Roszieg	0,702	0,733	0,031
Gunther	0,729	0,746	0,017
Gunther	0,708	0,729	0,021
Gunther	0,712	0,714	0,002
Gunther	0,702	0,713	0,011
Wee-mag	0,676	0,678	0,002
Wee-mag	0,663	0,677	0,014
Wee-mag	0,679	0,691	0,012
Wee-mag	0,696	0,700	0,004

CONCLUSIONS

In this paper a simulation based approach for calculating objective functions in metaheuristics procedures has been proposed. The formulation of objective function can in this way include all those performance parameters that are obtainable only as simulation outputs. This allows minimizing the difference between the target of the algorithm and the target that have to be reached in reality.

The procedure has been used to solve a generalized inspection line balancing problem by the use of tabu-search algorithm, but is extendable to all those security problems that can be approached by a metaheuristic procedure.

The experiment showed the effectiveness of the proposed technique.

REFERENCES

R.G. Askin, M. Zhou, 1997. A parallel station heuristic for the mixed-model production line balancing problem, *International Journal of Production Research*, 35, 3095-3105.

C. Becker, A. Scholl, 2003. A survey on problems and methods in generalized assembly line balancing, *Jenaer Schriften zur Wirtschaftswissenschaft 21/03*, FSU Jena.

J. Bukchin, 1998. A comparative study of performance measures for throughput of a mixed model assembly line in a JIT environment. *International Journal of Production Research*, 36, 2669-2685.

W.C. Chiang, 1998. The application of a tabu search metaheuristic to the assembly line balancing problem. *Annals of Operations Research*, 77, 209-227.

F. Glover, 1989a. Tabu search, Part I, *ORSA Journal on Computing*, 1, 190-206.

F. Glover, 1989b. Tabu search, Part II, *ORSA Journal on Computing*, 2, 4-32.

K. Helsgaun, 2004. Discrete event simulation in Java, Department of Computer Science, Roskilde University, Denmark.

Kelly Leone, Rongfang (Rachel) Liu, 2005. The key design parameters of checked baggage security screening systems in airports, *Journal of Air Transport Management*, 11, 69-78.

P.R. McMullen, G.V. Frazier, 1997. A heuristic for solving mixed-model line balancing problems with stochastic task durations and parallel stations. *International Journal of Production Economics*, 51, 177-190.

P.R. McMullen, G.V. Frazier, 1998. Using simulated annealing to solve a multiobjective assembly line balancing problem with parallel workstations". *International Journal of Production Research*, vol. 36, no. 10, 2717-2741.

P.R. McMullen, P. Tarasewich, 2003. "Using ant techniques to solve the assembly line balancing problem", *IIE Transactions* 35, 605-617.

A. Sholl, 1993. Dataset of Sholl, http://www.assembly-line-balancing.de/.

A. Scholl, C. Becker, 2003. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *Jenaer Schriften zur Wirtschaftswissenschaft 20/03, FSU Jena*.

P.M. Vilarinho, A.S. Simaria, 2002. A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations, *International Journal of Production Research*, 40, 1405-1420.

AUTHOR BIOGRAPHIES

STEFANO SAETTA, Laurea degree in Electrical Engineering, is Associate Professor of Industrial Plants at the Faculty of Engineering of the University of Perugia (ITALY). His research activity is focused on simulation and optimization of the supply chain, industrial plant simulation, decision supporting tools and operation research. His email address is: stefano.saetta@unipg.it.

LORENZO TIACCI, Doctoral degree in Industrial Engineering, is post-doctoral fellow at the Department of Industrial Engineering of the University of Perugia (ITALY). His research interests enclose simulation tools for layout optimization, web-based simulation and discrete-event simulation. His expertise involve Modelling and Simulation in Engineering and Artificial Intelligence applications. His email is: lorenzo.tiacci@unipg.it.