IMPROVED PERFORMANCE IN PROCESS PLANT LAYOUT PROBLEMS USING SYMMETRY-BREAKING CONSTRAINTS

Joakim Westerlund[a] and Lazaros G. Papageorgiou[b*]
[a]Process Design Laboratory, Faculty of Chemical Engineering,
Åbo Akademi University, Turku 20500, Finland
[b] Centre for Process Systems Engineering, Department of Chemical Engineering,
UCL (University College London), London WC1E 7JE, UK

Abstract

In this paper, a number of symmetry-breaking (SB) constraints are considered to improve the performance and decrease computational effort needed in the solution of Process Plant Layout (PPL) Problems. The PPL problems are formulated as mixed-integer linear (MILP) models. In order to determine the efficiency of the symmetry-breaking formulations, test runs are carried out on three different problem sets. Both single-floor and multi-floor problems are examined.

Keywords

MILP, Process Plant Layout, Symmetry Breaking

Introduction

Plant layout plays an important role in engineering design of industrial facilities, Patsiatzis and Papageorgiou (2003). The PPL problem involves decisions concerning the spatial allocation of equipment items and required connections among them. The resulting mathematical models are combinatorial optimisation problems which usually require significant computational effort for their solution. It is evident that enhancement of the computational effort required to solve these problems is a topic of great relevance. Symmetries in form of multiple optimal solutions often consume additional CPU time in already tedious layout problems thus worsening the overall solution performance.

The PPL problems can be formulated as non-convex MINLP or as discretised MILP problems according to Patsiatzis and Papageorgiou (2002). A number of rectangular units are to be sited in a rectangular plant area using a limited number of floors. The objective involves minimisation of layout costs including connectivity costs and construction costs. The detailed PPL for the given problem is determined by resolving number of floors, land area, floor allocation of each item and detailed floor layout.

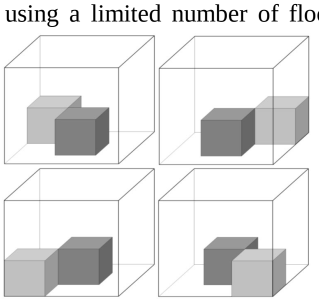

Figure 1. Example of symmetric layout solutions

Due to the geometry of the generalised PPL problem, symmetric layout solution alternatives (see figure 1) will be received for each problem implying the existence of multiple equivalent optimal solutions thus resulting in longer CPU time. Better efficiency in terms of CPU time can be obtained by breaking symmetric solutions.

Symmetry Breaking Constraints

In this paper, a set of different SB constraints are presented and empirically tested. Each constraint is tested on different kind of PPL problems and improvements of the performance is evaluated. SB constraint alt.1 is designed to break the symmetry in the layout using the following constraint for a pair of units *i* and *k*,

$$x_i + y_i - x_k - y_k \leq 0 \qquad (1)$$

The sum of the x- and y-values for the centroid of unit *i* is defined to be less or equal to the corresponding sum of item *k*. This constraint will lock unit *i*'s position in relation to unit *k*, thereby breaking the symmetry as shown in figure 2. SB constraint 1, applied to Facility Layout Problems, is presented in Westerlund and Castillo (2002).

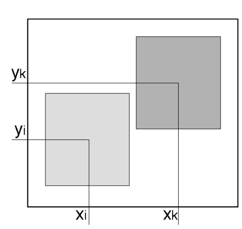

Figure 2. SB constraint Alt.1

---

* To whom all correspondence should be addressed

SB constraint alt.2 forces the centroid of unit *i* to be positioned in the lower left part of the whole facility area using the constraints in eq.(2) as illustrated in figure 3.

$$x_i - \tfrac{1}{2} X_{\max} \le 0 \;\&\; y_i - \tfrac{1}{2} Y_{\max} \le 0 \qquad (2)$$

In Eq.2 $X_{\max}$ is the length of the whole facility area in x-direction and $Y_{\max}$ is the length of the whole facility area in y-direction. This constraint will lock unit *i*'s position in relation to the whole facility area, thereby breaking the symmetry.

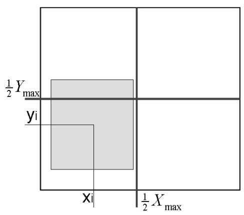

Figure 3. SB constraint alt.2

SB constraint alt.3 forces one of the non-overlapping binary variables to zero, together with constraints of alternative 1, equation (1). This procedure locks unit *i* in a compass point NE (North-East) of unit *k* as illustrated in figure 4. Unit *i* may also be locked in position NW, SE or SW of *k* by corresponding constraints. The constraints for alt.3 are,

$$x_i + y_i - x_k - y_k \ge 0 \;\&\; E1_{ik} = 0 \qquad (3)$$

The $E1_{ik}$ is one of two binary variables used to avoid overlapping between the two selected units *i* and *k*, Papageorgiou and Rotstein (1998). If $E1_{ik} = 0$ then, $x_i - x_k \ge \frac{l_i + l_k}{2}$ or $y_i - y_k \ge \frac{d_i + d_k}{2}$ according to the non-overlapping rules used in Patsiatzis and Papageorgiou (2002). SB constraint alt.3 is based on the symmetry-breaking considerations of Castillo and Westerlund (2002).

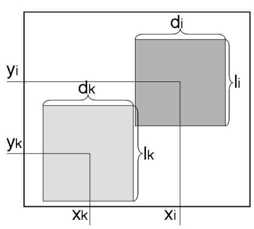

Figure 4. SB constraint alt.3

SB constraint alt.4 is a further developed version of alt.3 suitable for multi-floor cases,

$$x_i + y_i - x_k - y_k \ge \alpha \cdot Z_{ik} \;\&\; E1_{ik} = 0 \qquad (4)$$

Z_{ik} is a binary variable defining whether units *i* and *k* are positioned at the same floor or not. Z_{ik} need to be included in alt.4 in multi-floor cases to prevent exclusion of the global optimal solution in case units *i* and *k* are placed above each other at the same x- and y-variable values as shown in figure 5.

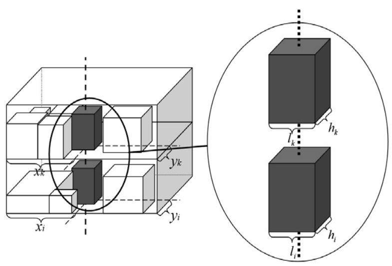

Figure 5. Multi-floor layout

Parameter α is defined as the sum of half the length, *l* or the depth, *d* (the ones that are smaller) of both the two units used in the constraint,

$$\alpha = \min(\tfrac{1}{2} l_i, \tfrac{1}{2} d_i) + \min(\tfrac{1}{2} l_k, \tfrac{1}{2} d_k) \qquad (5)$$

Units *i* and *k* may be selected in three different ways, as illustrated in examples ex2_1-ex2_10. In the first alternative (alt.4a) the two units with the highest connection costs are locked in relation to each other. The second alternative (alt.4b) uses the two smallest units and the third alternative (alt.4c) the two biggest units. A similar SB strategy as alternative 4, applied to facility layout problems, is presented in Sherali et al. (2002) where the two units having the largest flow are selected. In case of ties, the two units with the largest areas are selected. This corresponds to alt.4a and 4c in this paper.

Illustrative Examples

In this study, three different problem sets are solved to highlight the benefit of the SB constraints. Both single-floor and multi-floor problems are considered. The first problem set, ex1_1–ex1_10, consist of multi-floor problems with a total of five units to be optimally sited in a multi-floor process plant facility. The second set, ex2_1-ex2_10, also consists of multi-floor problems but with seven units to be sited in the multi-floor plant area. Both the first and the second set of problems are limited to use a maximum of three floors. The third problem set, SF_1 - SF_10 considers single-floor problems with seven units to be sited in the process plant area. The multi-floor example problems are based on the formulations of Patsiatzis and Papageorgiou (2002) and the single-floor problems are based on Papageorgiou and Rotstein (1998).

To get a clear picture of how the SB constraints affect both the solution quality and the computational effort required for solution, performance charts for each problem type are sketched. The first 10 problems are solved using two different discretisation grids. Problems solved using the first grid reaches solution faster while problems solved using the second one get better quality solutions as seen in figures 6 and 7. The two latter problem sets are solved using only a coarse discretisation grid. The performance charts (Figure 6, 7, 8 and 9) show the performance indicator versus the CPU time. The performance indicator displayed on the vertical axis is defined as,

$$\sum_{n=1}^{N} w_n \cdot \frac{1}{N} \quad \text{where} \quad w_n = 1 - \frac{J_n - J_n^*}{J_n^*} \qquad (6)$$

J_n is the solution in question for problem *n*, J_n^* is the global optimal solution and *N* the number of problems solved at CPU time *t*. The CPU time at each point is the CPU time required for solution of the considered problem the first point hence showing the fastest solution and the last point showing the slowest solution.

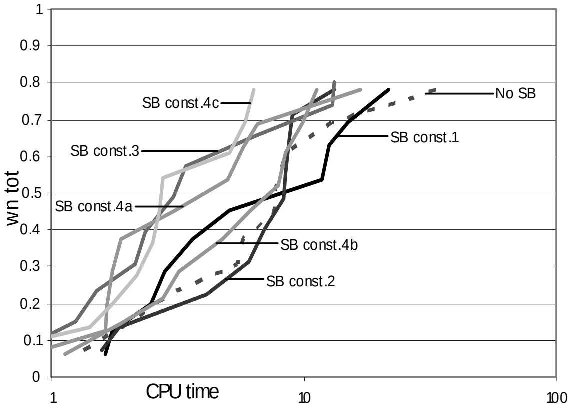

Figure 6. Performance charts for multi-floor examples 1_1 to 1_10 (Coarse grid)

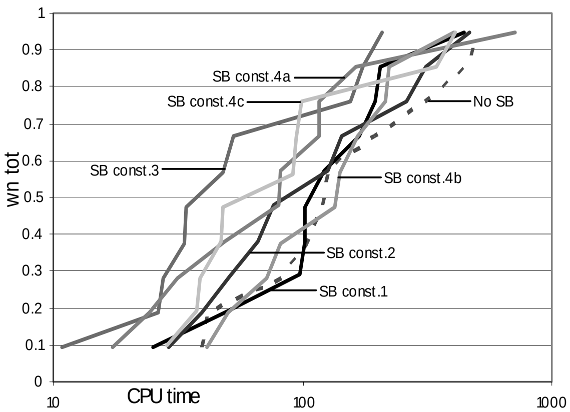

Figure 7. Performance charts for multi-floor examples 1_1 to 1_10 (Fine grid)

If a performance profile is on the left hand side of another, the first alternative solves the considered problems faster than the latter one. A performance profile ending up at a higher value than another indicates that the first alternative in average give better solutions than the latter one.

The endpoint of the performance profile in vertical direction gives the fractional average reached of the global optimum for all problems. The endpoint in time/horisontal direction indicates the CPU-time within which all problems considered were solved using the corresponding alternative. The endpoint alone should however not be seen as a performance indicator for the SB constraint in question. One slow test run might give misleading information of the SB constraint if the endpoint of the profile alone is seen as a characteristic for the average performance. This can for example be seen in fig.9 (Alt.1 & 2) and in fig.7 (Alt. 4a). The endpoint should be seen as the time required for all examples to be solved as it indicates the CPU time of the slowest example in the set.

The tables below (table 1-4) show both the CPU time and the solution quality of each problem in the three problem sets. The solution quality is shown as a percentage of the global optimum.

The global optimum of each problem is obtained with the global optimisation code GGPECP, Westerlund and Westerlund (2003) using the non-convex MINLP formulation in Patsiatzis and Papageorgiou (2002). The global optimum of each problem is shown in the last column in tables 1-4. As the objective in the problems is to minimise costs, values above 100% indicate that the global optimum is not reached. The solution quality values in tables 1-4 are only affected by the discretisation grid used and not by the SB constraints. The SB constraints solely affect the CPU time.

Table 1. Solution data for examples 1_1-1_10 (coarse grid)

	CPU time							Solution	Global
Ex	No SB	Alt.1	Alt.2	Alt.3	Alt.4a	Alt.4b	Alt.4c	Quality[%]	Optimum
1_1	5.5	2.8	6.0	3.4	5.7	11.3	5.9	113	72649
1_2	7.6	15.0	13.2	13.2	6.6	8.0	5.0	135	94452
1_3	3.0	3.7	7.0	2.4	1.7	8.4	2.8	111	174654
1_4	33.7	21.7	8.3	13.1	5.0	10.0	6.3	114	415672
1_5	8.6	5.1	9.0	6.4	3.1	6.2	2.2	121	77740
1_6	13.0	11.8	8.8	1.5	1.9	1.9	2.5	114	85462
1_7	1.9	1.8	1.9	0.8	0.8	1.1	1.4	138	31274
1_8	7.9	12.7	4.1	3.1	16.8	4.8	2.7	109	387644
1_9	1.4	2.5	1.6	2.2	1.7	3.2	0.6	128	53800
1_10	5.8	1.6	8.4	1.3	1.6	2.8	1.8	136	53532
Median	*6.7*	*4.4*	*7.6*	*2.7*	*2.5*	*5.5*	*2.6*		

Table 2. Solution data for examples 1_1-1_10 (fine grid)

	CPU time							Solution	Global
Ex	No SB	Alt.1	Alt.2	Alt.3	Alt.4a	Alt.4b	Alt.4c	Quality[%]	Optimum
1_1	120.4	98.0	76.4	34.3	80.8	135.0	99.3	101	72649
1_2	314.1	445.5	261.9	156.9	116.0	406.2	91.1	108	94452
1_3	126.1	122.0	51.2	53.1	81.5	166.4	28.6	103	174654
1_4	195.9	194.6	126.6	173.6	164.9	214.6	342.1	105	415672
1_5	103.6	102.5	66.0	10.9	116.2	72.2	39.2	107	77740
1_6	82.9	48.6	39.8	26.3	24.5	82.3	48.0	106	85462
1_7	41.5	25.2	29.1	27.8	17.2	50.3	47.3	105	31274
1_8	498.8	168.6	311.0	47.9	48.8	141.2	95.1	104	387644
1_9	458.8	204.9	465.3	210.0	716.5	223.4	413.6	108	53800
1_10	39.4	101.8	144.9	33.3	31.5	41.7	37.6	107	53532
Median	*123.2*	*112.2*	*101.5*	*41.1*	*81.1*	*138.1*	*69.5*		

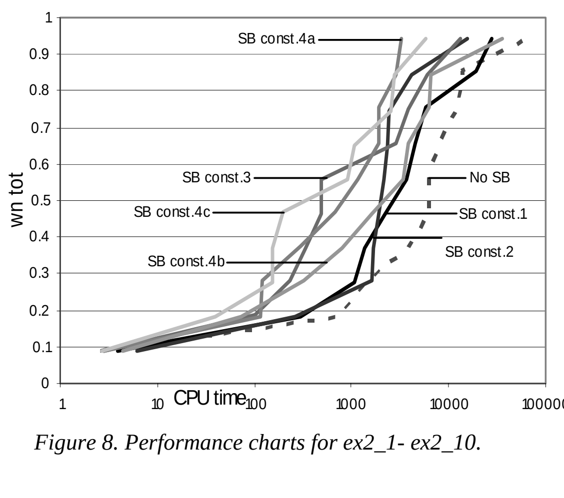

Figure 8. Performance charts for ex2_1- ex2_10.

Figures 6, 7, 8, and 9 show clear indications of the usefulness of some of the SB constraints. SB Alt.3 appears to work fine on most problems which can be seen by the fairly smooth shape of the curves. SB Alt.4a and 4c also show indications of even performance quality while alt.2 for example appears to work fine on some problems but even making the performance worse in other cases (see figure 6 and 9).

Effective SB constraints should show clear indications of even performance quality to assure that they do work properly in most cases. SB constraints Alt.4a and 4c are generally taken the most efficient alternatives used on both single-floor and multi-floor PPL problems and are therefore the ones recommended by the authors.

Table 3. Solution data for examples ex2_1 to ex2_10

	CPU time							Solution	Global
Ex.	No SB	Alt.1	Alt.2	Alt.3	Alt.4a	Alt.4b	Alt.4c	Quality[%]	Optimum
2_1	766.4	294.6	257.7	482.1	113.9	74.6	39.0	109	46742
2_2	3834.4	1083.8	1881.5	497.6	294.8	1628.2	153.7	106	68383
2_3	64875.2	27286.7	1665.6	336.4	1172.4	816.4	911.2	109	43330
2_4	1472.7	1357.2	2129.5	232.0	689.6	3439.4	156.7	107	63425
2_5	14074.5	2119.4	2366.0	3763.9	2895.2	6459.4	1054.3	108	67047
2_6	12517.1	19549.5	15366.8	6104.7	1919.8	35257.3	2515.5	103	49569
2_7	6379.7	4492.4	4089.0	2918.1	1932.9	6217.8	5841.6	101	93604
2_8	6311.0	3716.4	1642.2	100.8	119.1	328.1	198.0	102	54876
2_9	2.6	4.0	6.1	2.8	2.8	4.5	2.7	111	13068
2_10	8110.5	5724.8	2402.2	13396.2	3259.2	3803.6	2716.1	102	123585
Median	*6345.35*	*2917.87*	*2005.48*	*489.885*	*931*	*2533.8*	*554.6*		

Since no area cost is used in the single-floor problems, no discretisation of the plant area is needed and the global optimum of the examples is reached in every run. This is seen in figure 9 where all performance profiles reach a solution quality of 100%. The solution quality shown as a percentage of the global optimum is also shown in table4. The relatively smooth curves for SB constraint alternatives 4a, 4b and 4c seen in figure 9 indicate that the considered SB constraints work well on all the single-floor problems.

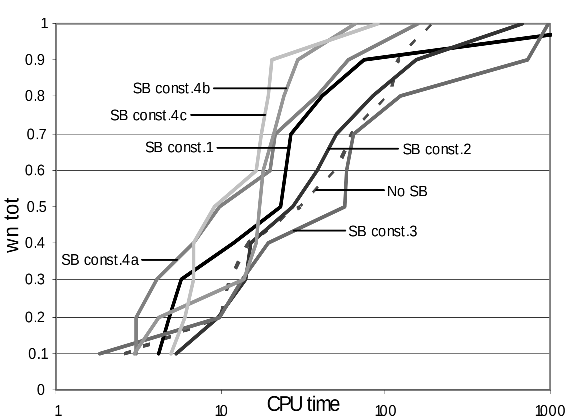

Figure 9. Performance charts for examples SF_1 –SF_10

Table 4. Solution data for examples SF_1- SF_10

	CPU time							Solution	Global
Ex.	No SB	Alt.1	Alt.2	Alt.3	Alt.4a	Alt.4b	Alt.4c	Quality[%]	Optimum
SF_1	11.4	5.8	5.3	733.0	3.0	4.2	6.8	100	9948.0
SF_2	31.1	22.8	27.5	9.9	20.1	21.1	6.9	100	11740
SF_3	119.8	73.4	50.1	13.6	4.1	17.9	20.5	100	5264
SF_4	9.9	4.8	15.3	64.7	6.8	13.7	6.0	100	13465
SF_5	63.2	26.6	84.5	56.2	38.1	29.1	16.5	100	11774
SF_6	194.0	41.4	38.3	966.7	59.6	64.9	17.7	100	8716
SF_7	50.0	3192.3	682.9	19.3	21.5	24.1	89.3	100	16769
SF_8	14.9	11.9	14.1	124.1	156.3	16.4	19.3	100	10049
SF_9	2.6	4.2	9.6	1.8	3.1	3.0	5.0	100	2074
SF_10	105.2	24.7	154.6	58.3	9.9	17.1	9.2	100	23953
Median	*40.5*	*23.8*	*32.9*	*57.2*	*15.0*	*17.5*	*12.8*		

The median CPU time shown in the last row of tables 1, 2, 3 and 4 is intended to complement the performance profiles in evaluating the performance of the SB constraints, giving each SB alternative an average performance indicator.

Conclusions

Symmetry breaking is an effective way of reducing computational efforts required to solve different kind of layout problems. The benefit of incorporating symmetry-breaking constraints within existing mathematical formulations for process plant layout has been illustrated through three different problem sets.

Acknowledgements

The first author gratefully acknowledges the financial support from TEKES, the National Technology Agency of Finland.

References

Castillo I. and Westerlund T. (2002). An ε-Accurate Model for Optimal Unequal-Area Facility Layout. *Operations Research Letters*, forthcoming.

Papageorgiou L.G. & Rotstein G.E. (1998). Continuous-Domain Mathematical Models for Optimal Process Plant Layout. *Ind. Eng. Chem.*, **37**, 3631-3639.

Patsiatzis D.I. & Papageorgiou L.G. (2002). Optimal Multi-Floor Process Plant Layout. *Computers and Chemical Engng.*, **26**, 575-583.

Patsiatzis D.I. & Papageorgiou L.G. (2003). Efficient Solution Approaches for the Multifloor Process Plant Layout Problem. *Ind.Eng. Chem.*, **42**, 811-824.

Sherali H.D., Fraticelli B.M. & Meller R.D. (2003). Enhanced Model Formulations for Optimal Facility Layout. *Operations Research*, **51**, 629-644.

Westerlund T. and Castillo I. (2002). Global Optimization of Facility Layout Problems with Varying Areas. *Operations Research*, Submitted.

Westerlund T. and Westerlund J. (2003). GGPECP – A Global Optimization MINLP Algorithm. *Chemical Engineering Transactions*, **3**, 1045-1050.