Optimal Integration of Departures and Arrivals in Terminal Airspace

Min Xue[*]

University of California at Santa Cruz, Moffett Field, CA 94035

Shannon Zelinski[†]

NASA Ames Research Center, Moffett Field, CA 94035

Coordination of operations with spatially and temporally shared resources, such as route segments, fixes, and runways, improves the efficiency of terminal airspace management. Problems in this category are, in general, computationally difficult compared to conventional scheduling problems. This paper presents a fast time algorithm formulation using a non-dominated sorting genetic algorithm (NSGA). It was first applied to a test problem introduced in existing literature. An experiment with a test problem showed that new methods can solve the 20 aircraft problem in fast time with a 65% or 440 second delay reduction using shared departure fixes. In order to test its application in a more realistic and complicated problem, the NSGA algorithm was applied to a problem in LAX terminal airspace, where interactions between 28% of LAX arrivals and 10% of LAX departures are resolved by spatial separation in current operations, which may introduce unnecessary delays. In this work, three types of separations – spatial, temporal, and hybrid separations – were formulated using the new algorithm. The hybrid separation combines both temporal and spatial separations. Results showed that although temporal separation achieved less delay than spatial separation with a small uncertainty buffer, spatial separation outperformed temporal separation when the uncertainty buffer was increased. Hybrid separation introduced much less delay than both spatial and temporal approaches. For a total of 15 interacting departures and arrivals, when compared to spatial separation, the delay reduction of hybrid

---

[*]Research scientist, University Affiliated Research Center. Mail Stop 210-8. AIAA senior member
[†]Research scientist, Aerospace High Density Operations Branch. Mail Stop 210-10. AIAA member

separation varied between 11% or 3.1 minutes and 64% or 10.7 minutes corresponding to an uncertainty buffer from 0 to 60 seconds. Furthermore, as a comparison with the NSGA algorithm, a First-Come-First-Serve based heuristic method was implemented for the hybrid separation. Experiments showed that the results from the NSGA algorithm have 9% to 42% less delay than the heuristic method with varied uncertainty buffer sizes.

I.   Introduction

In a metroplex or high density terminal operations, typically within 100 nautical miles of an airport or a major airport in a metroplex, resources, such as route segments, fixes, and runway, are normally spatially segregated in order to reduce interactions between different traffic flows and to guarantee separation between aircraft. Such separation may introduce unnecessary inefficiency due to lengthened routes or undesired altitude constraints, introducing integrated arrivals and/or departures with shared resources may help improve the efficiency.

In past years, in order to improve efficiency of terminal airspace operations, researchers treated the arrival and departure scheduling problems as separate ones. Many algorithms were developed, such as constrained position shifting (CPS),[1–3] CPS with dynamic programming (DP),[4] mixed integer linear programming (MILP),[5] and basic genetic algorithm (GA)[6] for optimizing schedules. Recently, spatial and temporal usage of shared resources started to draw researchers' attention. A couple of approaches were proposed to solve the optimal routing and scheduling problem. In 2009, Capozzi et. al[7] introduced a MILP formulation and applied it to coordinate SFO arrivals and SJC departures. The MILP formulation was found suitable for solving small scale problems but it required significant computational time if the number of flights was greater than 10. Later, the authors[8] further proposed a hybrid algorithm, which combined basic GA and MILP. The GA was used to solve a high level problem (route assignment and sequencing) while MILP was applied to the low level problems. Significant reduction in computational time was achieved when the algorithm was applied to a small problem, but further research is needed for solving realistic and complicated problems.

In this paper, a new formulation using a Non-dominated Sorting Genetic Algorithm (NSGA) was introduced because of its ability to handle multi-objective optimization and multiple constraints. The NSGA was demonstrated on a test problem first, then it was applied to an observed problem in LAX terminal airspace. Currently, interactions between LAX arrivals and departures are resolved by spatially segregating arrival and departure routes, which may introduce inefficiency. In this paper, three different separation approaches to the LAX problem were examined including the hybrid separation which combines temporal

and spatial separation. The temporal and hybrid separations were formulated and solved using the NSGA algorithm. Furthermore, a First-Come-First-Serve (FCFS) based heuristic method was applied to formulate the hybrid separation to compare with the NSGA algorithm.

In this paper, Section II introduces the NSGA algorithm. In section III, a test problem is set up to examine the performance of the new algorithm. In Section IV, the NSGA algorithm is applied to solve the interactions between arrivals and departures in LAX terminal airspace in three different ways. Analysis is then conducted to examine the benefits brought by integrating arrivals and departures spatially and temporally. Comparison between the NSGA and the heuristic method is also conducted. Section V discusses conclusions of the study.

II.   Methodology

In the terminal airspace, methods of integrating flights with shared resources include routing, sequencing, and scheduling. The objective is usually to minimize total delay time for a given set of flights while maintaining separation constraints and achieving proper sequence of landings. The problems are typically highly constrained due to aircraft separation requirements. According to Capozzi's work[8] the MILP formulation using the CPLEX solver requires a large amount of computational time, which might prohibit further application. In this work, a variation of GA is investigated.

GAs[9] have been quite successful in a great range of problems. These groups of algorithms are stochastic processes that model two natural phenomena: genetic inheritance and Darwinian evolution. Evolutionary operators include selection, crossover, and mutation. In the basic GA, the selection is based on the fitness functions of the population in the generation. Typical approaches to handle constraints are rejecting or penalizing infeasible individuals. The rejection of infeasible individuals is easy and popular, but it may get stuck when the feasible search space is not convex or the search space is highly constrained. Penalizing infeasible individuals relaxes the constraints, but it is hard to decide the penalty.

NSGA II,[10] the NSGA[11] variant used in this research was developed in recent years in order to improve GA's performance on multi-objective optimization and multiple constraint handling. Each individual has two attributes: fitness and errors. Fitness is calculated based on the objective, whereas errors are calculated if constraints are violated. Compared with the basic GA, the only change in NSGA is the selection process. Instead of fitness, the population is evaluated and ranked based on the ordering of their dominance (Pareto dominance) and is sorted into a hierarchy of subgroups. Assuming the objective is to minimize and the

constraint function g has to be nonnegative, individual A is dominated by individual B if:

$$\begin{cases} f_A > f_B, & \text{if } g_A \geq 0 \text{ and } g_B \geq 0, \text{ or, } g_A = g_B \\ g_A < g_B, & \text{if } g_A < 0 \text{ and } g_B < 0, \text{ or, } g_B > 0 \text{ and } g_A < 0 \end{cases} \tag{1}$$

Where f is the objective and g is the constraint value.

In order to estimate the density of solutions surrounding a particular solution in the population, an average distance between two neighboring solutions on either side of the particular solution along each objective is calculated and is termed as the crowding distance. Crowding distance between members of each sub-group introduces diversity among nondominated solutions.

Figure 1 demonstrates the selection procedure in NSGA II. At generation t, assuming P is the parent population and Q is the offspring population. Both P and Q have N individuals. They are first combined to a $2N$ population R_t, then individuals in R_t are sorted based on their Pareto dominance. The best non-dominated solutions are in set F_1. After calculating crowding distances of set F_1, if the size of F_1 is smaller than N, F_1 will be added into the new population P_{t+1}. Then, solutions from set F_2 are added to the new population, and so on. This process will continue unless the size of P_{t+1} exceeds N. In order to make sure that the size of P_{t+1} reaches N, the solutions from the next F_i will be partially chosen in terms of their crowding distances . The order $\succ$ in the flow is defined as Eqn. 2. After the selection process, the remaining processes are the same as basic GAs.

$$\begin{cases} A \succ B & \text{if } rank_A < rank_B, \text{ or } (rank_A = rank_B \text{ and } crowding\_distance_A > crowding\_distance_B) \end{cases} \tag{2}$$

In this work, bit strings were used to represent solutions. Therefore, the search space is discrete and it helps speed up the process without sacrificing much optimality. Usually, the population size was set to 800 and the maximum number of generations was defined to be 300. Because random initialization was used, any problem set in this study was usually run over three times and the best results were reported. All problems in this work were solved on a MacOS platform with 2x2.66GHz 6-Core Intel Xeon and 8GB RAM.

III. A Test Problem

To test the formulation using the NSGA algorithm, a test problem was rebuilt based on Capozzi's papers.[7,8] In this problem, two departure flows are assumed to come from two neighboring airports OAK and SJC, respectively. As shown in Fig. 2, the two departure
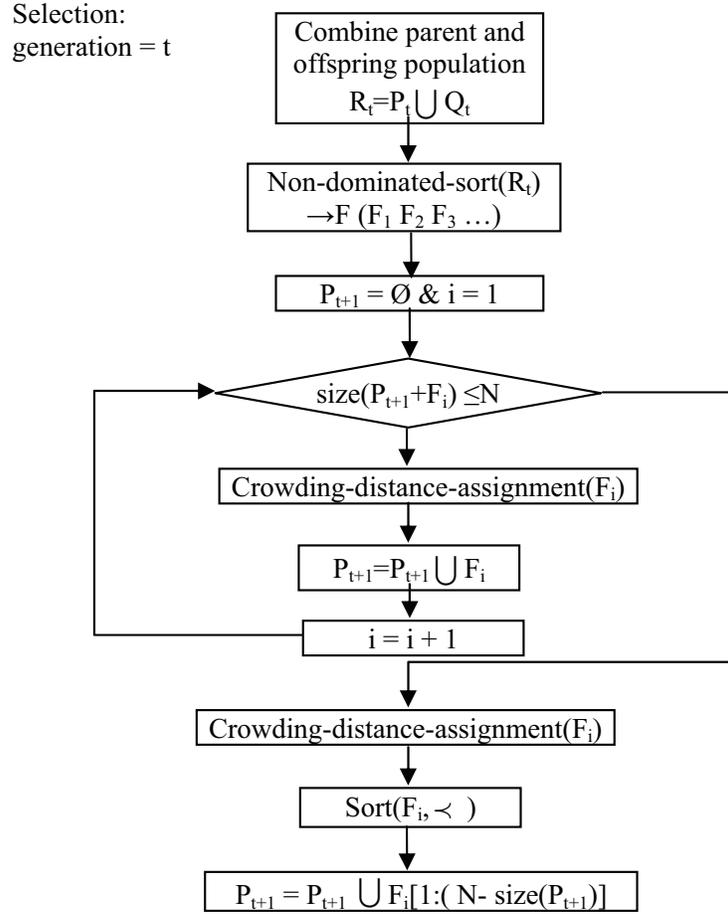
Figure 1. **NSGA II selection procedure**

fixes $W0$ and $W1$ are defined as shared resources. The distance for both $W1$ to OAK (route R_4) and $W0$ to OAK (route R_3) is 45 miles; The distance for both $W1$ to SJC (route R_2) and $W0$ to SJC (route R_1) is 56 miles. The flight departure schedule is described in Table 1, which is the same as the "phased peak demand" mentioned in Capozzi's paper.[7] Aircraft were assumed to be of the same type. The minimum speed of aircraft was defined to be 140 knots and the maximum speed was 180 knots. The minimum separation required at runway D_r was set to 3 nautical miles for all aircraft pairs and the minimum separation at departure fixes D_f was defined to be 4 nautical miles. The maximum delay was assumed to be 200 seconds. Flight time uncertainty was included using a time error (δ) of 60 seconds.

Two scenarios of fix usage were examined: segregated vs. shared. In the "segregated" case, only R_1 and R_4 are available, whereas in the "shared" case, all four routes are available. In this problem formulation, decision variables for each flight i are ground delays (d_i), aircraft speeds (v_i), and route options (r_i). For each flight, its time to exit a departure fix (tp_i) and its time to take off (tr_i) can be expressed as in Eqn. 3, where t_{i0} is the scheduled departure

Table 1. Scheduled departure time

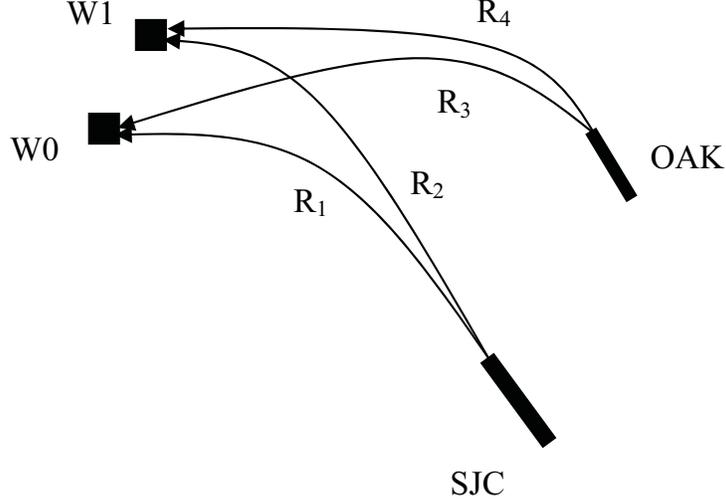| airport	schedule (seconds)								
SJC	0	300	600	900	1020	1120	1240	1400	1580	1780
OAK	0	120	220	340	520	720	1020	1320	1620	1920

Figure 2. Shared departure fixes for SJC and OAK

time as shown in Table 1. There are two routes for either SJC or OAK departures: $r_i = 0$ represents the default route (R_1 for SJC departures and R_4 for OAK departures) and $r_i = 1$ represents the second route (R_2 for SJC departures and R_3 for OAK departures). R_d means the default route and R_s is the second route. Variable v_i denotes the airspeed. Therefore, in NSGA, there are three genes (d_i, r_i, v_i) for each flight.

$$\begin{cases} tr_i = t_{i0} + d_i, \\ tp_i = tr_i + ((1 - r_i) \cdot R_d + r_i \cdot R_s)/v_i \end{cases} \tag{3}$$

$$f = \sum tp_i \tag{4}$$

The objective is minimizing the total time as in Eqn. 4. The constrains are: the separation at departure fixes (Eqn. 5) and separation at runways (Eqn. 6). The problem with 20 flights was modelled and solved without multiple windows that were used in MILP formulations.[7] The formulations in Eqn. 3 and 5 are built for the case of shared fixes. For the case of segregated fixes, r_i can be simply fixed at zero.

$$|tp_i - tp_j| \cdot [r_i \cdot r_j + (1 - r_i) \cdot (1 - r_j)] - D_f/v_k - \delta \geq 0, \quad (i \neq j, \ if \ tp_i > tp_j, k = j \ otherwise \ k = i) \tag{5}$$

$$|tr_i - tr_j| - D_r/v_k - \delta \geq 0, \quad (i \neq j, \ if \ tp_i > tp_j, k = j \ otherwise \ k = i) \tag{6}$$

Table 2 shows the resulting delays with segregated and shared fixes, respectively. A reduction of 65% was achieved by shared departure fixes. The total delay saving was 440 seconds (from 677 seconds to 237 seconds) over segregated fixes. Among them, 516 seconds were saved from ground delay with increased airborne delay of 76 seconds. Makespans were the same because it was constrained purely by flight schedule in this problem. Figure 3(a) and 3(b) presented the results for segregated and shared fixes, respectively. The vertical axes are way points. The times of SJCRWY and OAKRWY represent the scheduled departure times, while the times of SJC and OAK are departure times computed by the algorithm. The differences denotes the suggested delays. Note that many SJC departures used $W1$ which wasn't available to them in the "segregated" case and six out of ten OAK departures used $W0$ which wasn't available to them in the "segregated" case. Sharing departure fixes provides flexibility in route options, and the departure fixes can therefore be fully used to improve operation efficiency. This test showed that the new formulation with the NSGA performs well. It solved the 20 flight problem in around 30 seconds without any parallelization. Because one of the strengths of GA-like algorithms is parallelization, significant reduction in computational time could be easily realized.

Table 2. Comparison of delays

	Ground delay(sec)	Airborne delay (sec)	Total delay (sec)	Makespan (sec)
Segregated	645	32	677	2900
Shared	129	108	237	2900

IV.   Case Study: LAX Terminal Airspace

The interactions between arrivals and departures in LAX terminal airspace was identified to be a potential scheduling problem that could be solved more efficiently than current procedures. This section describes the model, method, and analysis for this problem.
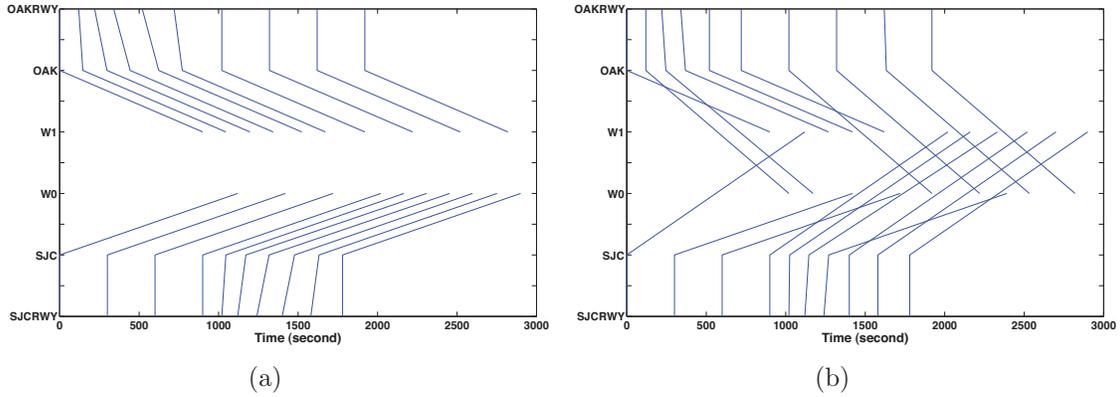
**Figure 3. Schedules for (a) segregated departure fixes and (b) shared departure fixes**

A. Description

According to the Standard Terminal Arrival Routes (STARs) of Los Angeles terminal airspace, arrivals to Los Angeles airport (LAX) from the North are required to take procedure SADDE6, which is to fly from Fillmore(FIM) to Santa Monica(SMO) via SYMON and SADDE fixes. Based on the Standard Instrument Departures (SIDs), LAX departures to the North need to follow procedure CASTA2, which is to take off from Runway 24R to WPT1 [a] through NAANC and GHART (see Fig. 4). In order to spatially segregate these two flows, arrival flights from FIM are required to maintain their flight altitudes above 12,000 feet at Fix GHART, while departures have to keep theirs at or below 9,000 feet at the same fix.

To resolve potential conflicts, departures and arrivals have to fly longer-than-necessary distances and constrain their climb and descent altitudes. If there was no arrival flow, departure flights could fly directly to WPT1 through WPT2 with a distance of 16.25 nmi as shown in Fig. 4 instead of the 24.69 nmi CASTA2 departure. And if there was no departure flow, arrivals from FIM could utilize the route of FIM-WPT1-SMO with a distance of 29.69 nmi instead of the 35.46 nmi SADDE6 approach. Direct routes could save approximately 60 seconds for an arrival flight and 120 seconds for a departure flight. It should also be noted that the extra inefficiency caused by undesired altitudes hasn't been taken into account.

As studied by Timar,[12] approximately 28.1% of LAX arrivals use the SADDE6 procedure and 10.4% of LAX departures use the CASTA2 procedure. In a typical day, this can be translated to 220 arrival flights and 80 departure flights. The total cost or delay in a day due to the arrival departure interaction is approximately 380 minutes with spatially segregated routes in the SIDs and STARs. Could it be improved? The following study addresses this using three different methods: spatial, temporal and hybrid separation. Spatial separation

---

[a]Points WPT1 and WPT2 are made up for simplicity.
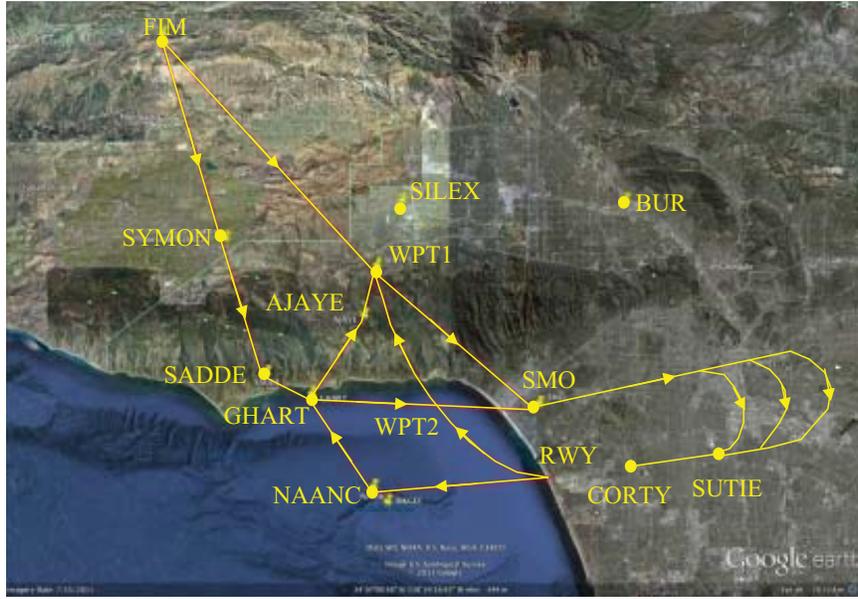
Figure 4. Interactions between SADDE arrivals and CASTA departures

uses the same strategy as in SIDs and STARs. Temporal separation utilizes the direct routes with conflicts resolved solely with temporal controls. As in Fig. 4, the direct route for departures is RWY-WPT2-WPT1, and for FIM arrivals it is FIM-WPT1-SMO-SUTIE. Hybrid separation applies both temporal and spatial separation.

B. Modeling

Three flows are taken into account in this work: arrivals from FIM (Set A_1), departures from Runway 24L(Set D), and arrival flights from the East towards SUTIE(Set A_2). Table 3 shows scheduled arrival times (t_{i0}) at FIM, RWY, and SUTIE, respectively. Historical traffic schedules between 18:30 pm to 19pm (UTC time) on March 5, 2010 were used as a reference for generating the schedules. There are a total of 15 flights including five arrivals from FIM, six arrivals from east of LAX, and four departures from Runway 24L. In this work, flights were assumed to be the same type.

1. Decision Variables

Assume that route R_1 refers to RWY-WPT2-WPT1 (direct route for departures), R_2 represents RWY-NAANC-GHART-AJAYE-WPT1 (lengthened route for departures), R_3 denotes FIM-WPT1-SMO-SUTIE (direct route for arrivals), and R_4 is the route of FIM-SADDE-GHART-WPT2-SMO-SUTIE (lengthened route for arrivals). In the formulation of hybrid separation, four design variables were defined for each arrival flight in Set A_1:

- $d1_i$ - The delay at or before FIM.

Table 3. Scheduled arrival times

Order	FIM (sec)	RWY (sec)	SUTIE (sec)
1	0	30	430
2	135	298	671
3	263	540	1070
4	860	1240	1210
5	1230	NA	1376
6	NA	NA	1780

- r_i - If $r_i = 0$, the direct route R_3 will be chosen, otherwise, R_4 is selected.

- v_i - The aircraft speed between FIM and WPT1 when flying the direct route or the speed between FIM and WPT2 if flying the indirect route.

- $d2_i$ - The delay at or before SUTIE to ensure separation at SUTIE.

For a departure flight in set D, three decision variables were defined:

- d_i - The delay before departure.

- r_i - If $r_i = 0$, the direct route R_1 will be chosen, otherwise R_2 is selected.

- v_i - The speed from departure to WPT1.

Only one decision variable exists for an arrival flight in Set A_2:

- d_i - The delay time at or before SUTIE to ensure separation with A_1 at SUTIE.

In the case of temporal separation, route options (r_i) in Set D and Set A_1 are fixed at zero so both departures and arrivals take direct routes. The only way to meet the separation requirements is to use time control. In the study, two scenarios were set up. In scenario one, no aircraft is allowed to arrive/depart early or speed up. In scenario two, aircraft are allowed to arrive/depart early or speed up up to 30 seconds.

2. Constraints

Eqn. 7 shows the expression for FIM arrivals. Let R'_3 denote route FIM-WPT1 and R'_4 represents route FIM-SADDE-WPT2, which are partial routes of R_3 and R_4, respectively. Variable $L_{R'_i}$ represents the length or distance of route R'_i. Variable $t_{FIM(FIM,i)}$ is defined to be the time when flight i arrives at FIM. Variable $t_{FIM(WPT,i)}$ denotes the arrival time of flight i at WPT1 if a direct route is chosen, or the arrival time of flight i at WPT2 if

the indirect route is selected. Variable $t_{FIM(SUTIE,i)}$ refers to the arrival time of flight i at SUTIE. The minimum travel time between WPT1 to SUTIE is defined to be 290 seconds. When R_4 is activated, the minimum travel time between WPT2 to SUTIE is set to 220 seconds.

$$\begin{cases} t_{FIM(FIM,i)} = t_{i0} + d1_i \\ t_{FIM(WPT,i)} = t_{FIM(FIM,i)} + [(1 - r_i) \cdot L_{R_3'} + r_i \cdot L_{R_4'}]/v_i \\ t_{FIM(SUTIE,i)} = t_{FIM(WPT,i)} + (1 - r_i) \cdot (d2_i + 290) + r_i \cdot (d2_i + 220) \end{cases} \tag{7}$$

Eqn. 8 shows the expression for departures, where $t_{DEP(RWY,j)}$ represents the time flight j departs from RWY, variable $t_{DEP(WPT2,j)}$ denotes the time flight j arrives at WPT2, and R_1' refers to the route RWY-WPT2. Variable $t_{DEP(WPT1,j)}$ represents the time flight j arrives at WPT1.

$$\begin{cases} t_{DEP(RWY,j)} = t_{i0} + d_j \\ t_{DEP(WPT2,j)} = t_{DEP(RWY,j)} + (1 - r_j) \cdot L_{R_1'}/v_j \\ t_{DEP(WPT1,j)} = t_{DEP(RWY,j)} + [(1 - r_j) \cdot L_{R_1} + r_j \cdot L_{R_2}]/v_j \end{cases} \tag{8}$$

Eqn. 9 presents the expression for A_2 arrivals with simply one decision variable.

$$t_{SUTIE(SUTIE,k)} = t_{i0} + d_k \tag{9}$$

Separation constraints were applied at fixes that could have potential violations, such as FIM, RWY, WPT1, WPT2, and SUTIE. Separation requirements were 3 nmi at the runway and 4 nmi elsewhere. As in the previous section, an uncertainty buffer of δ was added in the separation constraints for a sensitivity study.

3. Objective

The objective is to minimize the sum of exit times, as shown in Eqn. 10. For departures it is the time when a flight leaves the waypoint WPT1. For arrivals, it is the time when a flight reaches waypoint SUTIE.

$$J = \sum_{i,j,k} t_{FIM(SUTIE,i)} + t_{DEP(WPT1,j)} + t_{SUTIE(SUTIE,k)} \tag{10}$$

C. Results

In this section, results using three different separation methods are presented and compared. Different buffers are set up to deal with uncertainty and the impacts are studied. In addition,

two scenarios are defined: only delays are considered in the first one, while early arrivals are allowed in the second case. For any flight, its unimpeded flight time (fly via direct route without any consideration of separation from other flights) is treated as a baseline. Beyond that, any extra flight time will be called delay.

1. Comparison of separation methods

This section compares separation methods when uncertainty buffers were defined to be zero. In the case of spatial separation, because the indirect route was the only option, there exists associated extra flight time of 771 seconds. Because all flights were assumed to be the same aircraft type, the total delay can be computed manually. Including route-caused delay the total delay is 1,001 seconds as shown in the Table 4. It can be seen that without uncertainty buffers the delay with hybrid separation was 357 seconds – a reduction of 64% or 10.7 minutes compared to the 1,001 second delay with spatial separation. While in this case, the temporal separation also achieved much less delay than the spatial separation. The reduction was 59% or 9.8 minutes.

Table 4. Total delay with different separation methods

uncertainty buffer	Spatial	Temporal	Hybrid
0 s	1,001s	413 s	357 s
30 s	1,163 s	776 s	759 s
60 s	1,673 s	1,808 s	1,490 s

Tables 5, 6 and 7 show individual flight results for sets A_1, D_1, and A_2, respectively, when the hybrid separation is applied. Table 5 provides the results for arrivals from FIM. It indicates that three of five arrival flights can make use of the direct route to reduce overall delay. In Table 6, all four departures flew their direct routes. Among the arrivals from the East, two flights were assigned some delays. The resulting delay under hybrid separation was reduced to 357 seconds as shown in Table 4.

2. Impact of uncertainty

The results of schedulers could be sensitive to the uncertainty of flight times. Robustness is required in actual operations. The easy and popular way to increase the robustness is to introduce an uncertainty buffer for flight times. As a trade-off, adding buffers causes additional delays. In this study, the buffers of 30 and 60 seconds were applied. Table 4 shows the results. In the table, the temporal separation introduced much less delay than the spatial separation in the deterministic case, but when the uncertainty buffer increased

Table 5. Results for A_1 arrivals with hybrid separation and zero uncertainty

A_1 Arrivals	delay at FIM (s)	Route	Speed (kt)	delay at SUTIE (s)
Flight 1	0	*direct*	349	39
Flight 2	68	*indirect*	344	15
Flight 3	0	*indirect*	319	12
Flight 4	0	*direct*	350	0
Flight 5	0	*direct*	350	0

Table 6. Results for departures (D_1) with hybrid separation and zero uncertainty

Departures	delay before departure (s)	Route	Speed (kt)
Flight 1	0	*direct*	250
Flight 2	0	*direct*	250
Flight 3	8	*direct*	250
Flight 4	0	*direct*	250

Table 7. Results for A_2 arrivals with hybrid separation and zero uncertainty

A_2 Arrivals	delay before SUTIE (s)
Flight 1	0
Flight 2	0
Flight 3	8
Flight 4	0
Flight 5	62
Flight 6	4

to 60 seconds, the temporal separation caused more delay than spatial separation. This showed that temporal separation was sensitive to the uncertainty buffer and corresponding schedules might be undesired in actual operations. The hybrid approach generated the least delay compared with the other two approaches, although the reduction decreased to 183 seconds when the buffer was 60 seconds. Results in Table 4 show the trade-off between buffer size and delays. In order to find out the best balance, an uncertainty study is required for future work.

Figures 5 and 6 show the time lines for all metering points when the buffer is zero and 30 seconds, respectively, where the hybrid approach was applied. SUT stands for SUTIE
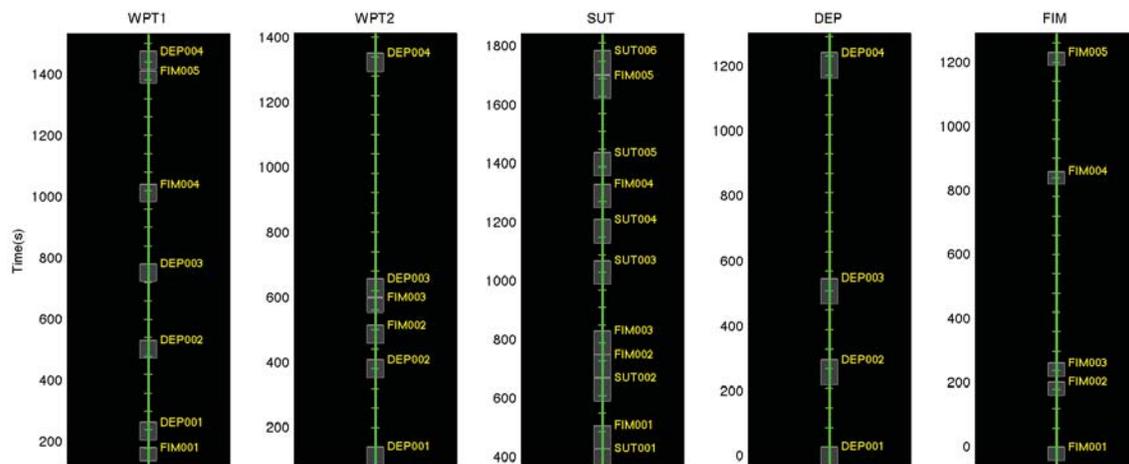
Figure 5. Time Line with hybrid separation and buffer = 0 seconds

for simplicity. Each flight has a safe zone shown as a grey box in front of its arrival time. These safe zones can be packed in Fig. 5, but they are well separated by a 30 second buffer in Fig. 6. Note how the flight loading of WPT1 and WPT2 changes significantly between 0 and 30 seconds. When the buffer is zero, FIM001, FIM004, and FIM005 are proposed to fly the direct route, but when the buffer increases to 30 seconds, the other FIM arrivals – FIM002 and FIM003 – are proposed to take the short cut.
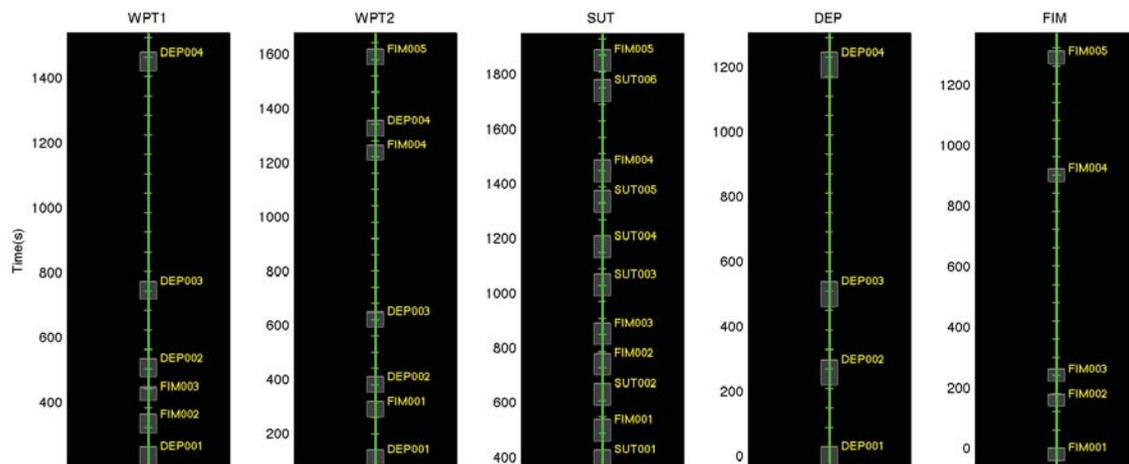

Figure 6. Time Line with hybrid separation and buffer = 30 seconds

3. Impact of early arrivals

Allowing early arrival or speeding up increases flexibility and reduces delay. In the previous cases, no early arrival or speeding up was allowed. In this section, early arrival is allowed up to 30 seconds. The results are presented in Table 8. For spatial separation, the delay

was simply calculated by shifting each aircraft 30 seconds earlier than the previous case. The temporal and hybrid separations were solved using the NSGA algorithm as in previous sections. As shown in the table, when the uncertainty buffer was zero, total delays caused by both temporal and hybrid separation were negative, which means on average aircraft arrived early. The delay reduction from spatial separation to hybrid separation was 12.5 minutes when the uncertainty buffer is zero and the reduction decreased to 3 minutes when the buffer increased to 60 seconds. On the other hand, the uncertainty buffer in flight time still plays an important role. When the buffer increased, the delay required increased quickly, especially for the temporal separation.

Table 8. Total delay when early arrival is allowed

uncertainty buffer	Spatial	Temporal	Hybrid
0 s	551 s	-170 s	-195 s
30 s	713 s	393 s	309 s
60 s	1,223 s	1,338 s	1,041 s

4. NSGA algorithm v.s Heuristic method

The performance of the NSGA algorithm is compared with that of a FCFS based heuristic method. Because this problem involves route option and multiple scheduling points, the rules of FCFS may not be straightforward and have to be clarified:

- The estimated or scheduled entering times are used as references for setting up priority. For FIM arrivals, the entering times are the arrival times to fix FIM. For departures, they are the estimated times of leaving RWY. And the estimated SUTIE arrival times are used as references for arrivals from the East. Each flight decides its route based on the FCFS rule in the order of their entering times. The route that causes the lowest delay at the time will be chosen.

- The conflicting flights in metering points WPT1, WPT2, and SUTIE are resolved based on the estimated arrival times, not their entering times. For instance, assume flight A has an earlier entering time than flight B, but the estimated arrival time of flight A at fix F is later than flight B. If flight A would conflict with flight B at fix F, then flight A would be delayed before it reaches fix F. With this rule, the arrival sequences at these metering points are actually allowed to be changed. This may be different from the strict FCFS rule.

- No flight should be delayed more than M seconds at any fix. M is set to 200 in this work.

- The order of flights in the same flow should be kept, e.g. FIM arrivals, SUTIE arrivals, and departures.

In the NSGA algorithm, the resolution of delay was approximately five to ten seconds. Table 9 presents the results using the heuristic method with a delay resolution of one second. When the uncertainty buffer is zero, the results generated by the NSGA algorithm can save about 42% over the heuristics method even though the latter has high resolution. When the buffer size increases, that saving was reduced to 20% or 9%, which was probably due to the decreased solution space. Overall, the proposed NSGA algorithm outperformed the heuristic method with a great difference. Unlike conventional scheduling problems, the optimization method showed greater advantage over heuristics due to the complicated solution space. On the other hand, it is also noticed that when the buffer size is large enough, the benefit of integration of arrivals and departures could disappear and the heuristic method becomes a good choice due to its computational easiness.

Table 9. Delay with hybrid separation using different methods

uncertainty buffer	NSGA algorithm	Heuristic	Difference
0 s	357 s	611 s	42%
30 s	758 s	950 s	20%
60 s	1,490 s	1,638 s	9%

V. Conclusion

The integration of departures and arrivals seems promising in improving operational efficiency in terminal airspace. The problem combines routing and scheduling problems, which further challenges the MILP solver. This work introduced a variation of genetic algorithm – NSGA. The NSGA was used because it is better than basic GA in handling constraints. Results with a test problem showed that the new formulation with NSGA can solve the problem in a fast time fashion.

A potential application of integrated arrivals and departures was identified in LAX terminal airspace. The arrival and departure route structures were modelled with three different strategies: spatial separation, temporal separation, and hybrid separation. A problem was set up based on a historical traffic schedule with a total of 15 flights and three flows included. The results showed that although the temporal separation introduced much less

delay than the spatial separation, it caused more delay than the latter when the uncertainty buffer increased to 60 seconds. The hybrid separation outperformed both the temporal and the spatial separations: It reduced unnecessary delay by 64% or 10.7 minutes if no early arrival/departure or speeding up was allowed; And if early arrival/departure or speeding up was allowed, the saving increased to 12.5 minutes. Compared with a FCFS based heuristic method, the schedules produced by the NSGA saved flight time up to 42%, which showed greater advantage over FCFS than typically seen in conventional scheduling problems. Overall, this study showed that it is promising to improve operation efficiency in LAX terminal airspace by integrating departures and arrivals using hybrid separation with the NSGA algorithm. Apparently, such efficiency may vary with aircraft departure and arrival schedules. The proposed method can be applied in a fast time fashion to decide if benefit exists and how to quantify it. Therefore, it can help decision makers to operate properly.

In order to achieve this goal, an analysis needs to be completed for the uncertainty in the schedules. In the future work, such uncertainty analysis will be conducted. The robustness of the benefits and controllers' workload will be examined by imposing flight time perturbations. The balancing point for the trade-off between robustness and delay will be studied. The schedulers using different separation approaches and different algorithms will be investigated and compared.

References

[1]Dear, R. G., "The Dynamic Scheduling of Aircraft in the Near Terminal Area," Tech. Rep. Flight Transportation Laboratory Report R76-9, MIT, September 1976.

[2]Dear, R. G. and Sherif, Y. S., "An Algorithm For Computer Assisted Sequencing and Scheduling of Terminal Operations," *Transportation Research A*, Vol. 25A, No. 2/3, 1991.

[3]Neuman, F. and Erzberger, H., "Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Spacing," Tech. Rep. NASA Technical Memorandum 103880, October 1991.

[4]Balakrishnan, H. and Chandran, B., "Scheduling Aircraft Landings Under Constrained Position Shifting," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 21-24 August 2006.

[5]Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D., "Scheduling Aircraft Landings - The Static Case," *Transportation Science*, Vol. 34, No. 2, 2000.

[6]Kupfer, M., "Scheduling Aircraft Landings to Closely Spaced Parallel Runways," *The Eighth USA/Europe Air Traffic Management Research and Development Seminar*, Napa, CA, June 2009.

[7]Capozzi, B. J. and Atkins, S. C., "Towards Optimal Routing and Scheduling of Metroplex Operations," *9th AIAA Aviation Technology, Integration, and Operations Conference(ATIO)*, Hilton Head, South Carolina, 21-23 September 2009.

[8]Capozzi, B. J. and Atkins, S. C., "A Hybrid Optimization Approach to Air Traffic Management for Metroplex Operations," *10th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Fort Worth, Texas, 13-15 September 2010.

[9]Michaelwicz, Z., *Genetic Algorithms + Data Structures = Evolution Programs, Third Edition*, Springer-Verlag Berlin Heidelberg, 1996.

[10]Deb, K., Pretap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions On Evolutionary Computation*, Vol. 6, No. 2, 2002.

[11]Srinivas, N. and Deb, K., "Multi-Objective Function Optimization using Non-dominated Sorting Genetic Algorithms," *Evolutionary Computation*, Vol. 2, No. 3, 1995.

[12]Timar, S. D., Nagle, G., Saraf, A., Yu, P., Hunter, P., Trapani, A., and Johnson, N., "Super Density Operations Airspace Modeling for the Southern California Metroplex," *AIAA Modeling and Simulation Technology Conference*, Portland, Oregon, 08-11 August 2011.